

165

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-158971

(43)Date of publication of application : 31.05.2002

-----  
(51)Int.Cl. H04N 5/92

G11B 20/12

G11B 27/00

H04N 5/91

-----  
(21)Application number : 2001-089959 (71)Applicant : SONY CORP

(22)Date of filing : 27.03.2001 (72)Inventor : KATO MOTOKI  
HAMADA TOSHIYA

-----  
(30)Priority

Priority number : 2000183770

2000268042

Priority date : 21.04.2000

05.09.2000

Priority country : JP

JP

-----  
(54) INFORMATION PROCESSOR AND PROCESSING METHOD, AND RECORDING  
MEDIUM THEREFOR, AND PROGRAM AND ITS RECORDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To retrieve a specified mark quickly.

SOLUTION: An AV stream, as an entity of contents, is managed by ClipInformation and reproduction of the AV stream is managed by PlayList. As the attribute information of the AV stream, address information on discontinuity points in the AV stream RSPN arrival time discontinuity, information on associating the time information and address information in the AV stream EP map, TU map, and time information on a characteristic image in the AV stream ClipMark are recorded in the ClipInformation.

\* NOTICES \*

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.\*\*\*\* shows the word which can not be translated.

3.In the drawings, any words are not translated.

---

## CLAIMS

---

[Claim(s)]

[Claim 1]An information processor which records said AV stream to a recording

medium with which an AV stream and Clip information which manages substance of said AV stream characterized by comprising the following, and PlayList information which manages reproduction of said AV stream are recorded.

A creating means in which encoded information in said AV stream generates start address information [ \*\*\*\* ] on the section, information which associates time information and address information in said AV stream, and time information of a characteristic picture in said AV stream.

A recording device recorded on said recording medium by making into said Clip information information generated by said creating means.

[Claim 2]Start address information [ \*\*\*\* / said encoded information ] on the section, The information processor according to claim 1, wherein it is a start address of an STC sequence or a program sequence, information which associates said time information and address information is EP\_map or TU\_map and time information of said characteristic picture is ClipMark.

[Claim 3]The information processor according to claim 1, wherein said recording device records further information about average value of a recording rate of said AV stream on said recording medium.

[Claim 4]The information processor according to claim 3, wherein information about said average value is TS\_average\_rate.

[Claim 5]The information processor according to claim 1, wherein said AV stream is a transport stream.

[Claim 6]The information processor according to claim 1, wherein start address information [ \*\*\*\* / encoded information in said AV stream ] on the section includes a start address of an STC sequence which is the section [ \*\*\*\* / a system time clock of said transport stream ].

[Claim 7]The information processor according to claim 1, wherein start address information [ \*\*\*\* / encoded information in said AV stream ] on the section includes a start address of a program sequence which is the section where the contents of a program of said transport stream are fixed.

[Claim 8]The information processor according to claim 1, wherein start address information [ \*\*\*\* / encoded information in said AV stream ] on the section includes a start address of the section [ \*\*\*\* / an arrival time based on the time of arrival of a transport packet of said transport stream ].

[Claim 9]The information processor according to claim 1, wherein information which associates time information and address information in said AV stream contains an

address and its presentation time stamp of I picture of said transport stream.

[Claim 10]The information processor according to claim 9, wherein it has further a compression means which compresses at least a part of information which associates time information and address information in said AV stream and said recording device records said information compressed by said compression means.

[Claim 11]The information processor comprising according to claim 1:

An arrival time based on the arrival time of a transport packet in information which associates time information and address information in said AV stream.

An address in said AV stream data of a transport packet corresponding to it.

[Claim 12]An account method of information processing of an information processor which records said AV stream to a recording medium with which an AV stream and Clip information which manages substance of said AV stream characterized by comprising the following, and PlayList information which manages reproduction of said AV stream are recorded.

A generation step in which encoded information in said AV stream generates start address information [ \*\*\*\* ] on the section, information which associates time information and address information in said AV stream, and time information of a characteristic picture in said AV stream.

A record step recorded on said recording medium by making into said Clip information information generated by said generation step.

[Claim 13]A program of an information processor which records said AV stream to a recording medium with which an AV stream and Clip information which manages substance of said AV stream characterized by comprising the following, and PlayList information which manages reproduction of said AV stream are recorded.

A generation step in which encoded information in said AV stream generates start address information [ \*\*\*\* ] on the section, information which associates time information and address information in said AV stream, and time information of a characteristic picture in said AV stream.

A record step recorded on said recording medium by making into said Clip information information generated by said generation step.

[Claim 14]Clip information which manages substance of an AV stream and said AV stream, To and a computer which controls an information processor which records said AV stream to a recording medium with which PlayList information which manages



reproduction of said AV stream is recorded. A generation step in which encoded information in said AV stream generates start address information [ \*\*\*\* ] on the section, information which associates time information and address information in said AV stream, and time information of a characteristic picture in said AV stream, A program which performs a record step recorded on said recording medium by making into said Clip information information generated by said generation step.

[Claim 15]An information processor which reproduces said AV stream from a recording medium with which an AV stream and Clip information which manages substance of said AV stream characterized by comprising the following, and PlayList information which manages reproduction of said AV stream are recorded.

A reproduction means in which encoded information in said AV stream reproduces start address information [ \*\*\*\* ] on the section, information which associates time information and address information in said AV stream, and time information of a characteristic picture in said AV stream as said Clip information.

A control means which controls an output of said AV stream based on said Clip information reproduced by said reproduction means.

[Claim 16]An information processing method of an information processor which reproduces said AV stream from a recording medium with which an AV stream and Clip information which manages substance of said AV stream characterized by comprising the following, and PlayList information which manages reproduction of said AV stream are recorded.

Regeneration steps in which encoded information in said AV stream reproduces start address information [ \*\*\*\* ] on the section, information which associates time information and address information in said AV stream, and time information of a characteristic picture in said AV stream as said Clip information.

A control step which controls an output of said AV stream based on said Clip information reproduced by processing of said regeneration steps.

[Claim 17]A program of an information processor which reproduces said AV stream from a recording medium with which an AV stream and Clip information which manages substance of said AV stream characterized by comprising the following, and PlayList information which manages reproduction of said AV stream are recorded. Regeneration steps in which encoded information in said AV stream reproduces start address information [ \*\*\*\* ] on the section, information which associates time information and address information in said AV stream, and time information of a

characteristic picture in said AV stream as said Clip information.

A control step which controls an output of said AV stream based on said Clip information reproduced by processing of said regeneration steps.

[Claim 18]Clip information which manages substance of an AV stream and said AV stream, And to a computer to control, an information processor which reproduces said AV stream from a recording medium with which PlayList information which manages reproduction of said AV stream is recorded as said Clip information, Regeneration steps in which encoded information in said AV stream reproduces start address information [ \*\*\*\* ] on the section, information which associates time information and address information in said AV stream, and time information of a characteristic picture in said AV stream, A program which performs a control step which controls an output of said AV stream based on said Clip information reproduced by processing of said regeneration steps.

[Claim 19]In a recording medium with which Clip information which manages substance of an AV stream and said AV stream, and PlayList information which manages reproduction of said AV stream are recorded, A recording medium, wherein start address information [ \*\*\*\* / encoded information in said AV stream ] on the section, information which associates time information and address information in said AV stream, and time information of a characteristic picture in said AV stream are recorded as said Clip information.

**\* NOTICES \***

**JPO and INPIT are not responsible for any damages caused by the use of this translation.**

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.\*\*\*\* shows the word which can not be translated.

3.In the drawings, any words are not translated.

---

## DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Field of the Invention]Especially this invention relates to the information processor and the method, recording medium, and program which record information, including the address information of I picture in an AV stream, an encoding parameter, change point information, a mark, etc., as a file about an information processor and a method, a recording medium, and a program.

[0002]

[Description of the Prior Art]In recent years, various kinds of optical discs are being proposed as a dismountable disk type recording medium from a recording and reproducing device. The optical disc in which such record is possible is proposed as several gigabytes of mass media.

The expectation as media which record AV (Audio Visual) signals, such as a video signal, is high.

As source (supply source) of the digital AV signal recorded on the optical disc in which this record is possible, there are CS digital satellite broadcasting and BS digital broadcasting, and the terrestrial television broadcasting of the digital system, etc. are proposed in the future.

[0003]Here, as for the digital video signal supplied from these source, it is common that graphical data compression is usually carried out by MPEG(Moving Picture Experts Group) 2 method. The recording rate peculiar to the device is provided in the recorder. By the conventional noncommercial image storage medium, if it is an analog recording method when recording the digital video signal of digital broadcasting, after decoding a digital video signal, a band limit will be carried out and it will record. Or if it is digital recording systems including MPEG1 Video, MPEG 2 Video, and DV method, after being decoded once, with a recording rate and a coding mode peculiar to the device, it will be re-encoded and will be recorded.

[0004]However, such a record method decodes the supplied bit stream once, and it is accompanied by degradation of image quality in order to record by performing band limit and re-encoding after that. When the transmission rate of the digital signal

inputted when the digital signal by which graphical data compression was carried out was recorded does not exceed the recording rate of a recording and reproducing device, decoding and the method of recording as it is, without re-encoding have least degradation of image quality in the supplied bit stream. However, when the transmission rate of the digital signal by which graphical data compression was carried out exceeds the recording rate of the disk as a recording medium, it is necessary to carry out re-encoding and to record so that a transmission rate may become below a maximum of the recording rate of a disk after decoding with a recording and reproducing device.

[0005]When the bit rate of the input digital signal is transmitted by the variable rate method fluctuated by time, Since a rotary head is fixed number of rotations, a recording rate stores data in a buffer once compared with the tape recording system which becomes a fixed rate, and the disc recording device which can do record burstily can use the capacity of a recording medium without futility.

[0006]As mentioned above, in the future which becomes in use, digital broadcasting is predicted that decoding and the recording and reproducing device which recorded without re-encoding and uses a disk as recording media are asked for a broadcasting signal like a data streamer with a digital signal.

[0007]

[Problem(s) to be Solved by the Invention]When reproducing the recording medium with which two or more data (for example, data of the program which comprises picture image data, voice data, etc.) is recorded with the device which was mentioned above, a user's random access and directions of special reproduction are received, Although processings, such as determination of the reading position of the AV stream from a recording medium and decoding of a stream, had to be performed promptly, the technical problem that such processing could not be made prompt occurred as the data volume recorded on a recording medium increased.

[0008]By making this invention in view of such a situation, and recording information, including the address information of I picture in an AV stream, an encoding parameter, change point information, a mark, etc., as a file, It enables it to perform promptly the determination and decoding processing of a reading position of an AV stream, and aims at enabling it to search a predetermined mark promptly especially.

[0009]

[Means for Solving the Problem]This invention is characterized by the 1st information processor comprising the following.

A creating means in which encoded information in an AV stream generates start

address information [ \*\*\*\* ] on the section, information which associates time information and address information in an AV stream, and time information of a characteristic picture in an AV stream.

A recording device recorded on a recording medium by making into Clip information information generated by creating means.

[0010]Start address information [ \*\*\*\* / said encoded information ] on the section, It is a start address of an STC sequence or a program sequence, and information which associates time information and address information is EP\_map or TU\_map, and time information of a characteristic picture can be set to ClipMark.

[0011]Said recording device can record further information about average value of a recording rate of an AV stream on a recording medium.

[0012]Information about said average value can be made into TS\_average\_rate.

[0013]Said AV stream can be made into a transport stream.

[0014]The start address information [ \*\*\*\* / encoded information in said AV stream ] on the section can include a start address of an STC sequence which is the section [ \*\*\*\* / a system time clock of a transport stream ].

[0015]The start address information [ \*\*\*\* / encoded information in said AV stream ] on the section can include a start address of a program sequence which is the section where the contents of a program of a transport stream are fixed.

[0016]The start address information [ \*\*\*\* / encoded information in said AV stream ] on the section can include a start address of the section [ \*\*\*\* / an arrival time based on the time of arrival of a transport packet of a transport stream ].

[0017]The information which associates time information and address information in said AV stream can contain an address and its presentation time stamp of I picture of a transport stream.

[0018]It can have further a compression means which compresses at least a part of information which associates time information and address information in said AV stream, and the recording device can record information compressed by compression means.

[0019]The information which associates time information and address information in said AV stream can include an address in AV stream data of an arrival time based on the arrival time of a transport packet, and a transport packet corresponding to it.

[0020]This invention is characterized by the 1st information processing method comprising the following.

A generation step in which encoded information in an AV stream generates start

address information [ \*\*\*\* ] on the section, information which associates time information and address information in an AV stream, and time information of a characteristic picture in an AV stream.

A record step recorded on a recording medium by making into Clip information information generated by generation step.

[0021]This invention is characterized by a program of the 1st recording medium comprising the following.

A generation step in which encoded information in an AV stream generates start address information [ \*\*\*\* ] on the section, information which associates time information and address information in an AV stream, and time information of a characteristic picture in an AV stream.

A record step recorded on a recording medium by making into Clip information information generated by generation step.

[0022]Start address information [ \*\*\*\* / the 1st program of this invention / encoded information in an AV stream ] on the section, A generation step which generates information which associates time information and address information in an AV stream, and time information of a characteristic picture in an AV stream, and a record step recorded on a recording medium by making into Clip information information generated by generation step are performed.

[0023]This invention is characterized by the 2nd information processor comprising the following.

A reproduction means in which encoded information in an AV stream reproduces start address information [ \*\*\*\* ] on the section, information which associates time information and address information in an AV stream, and time information of a characteristic picture in an AV stream as Clip information.

A control means which controls an output of an AV stream based on Clip information reproduced by reproduction means.

[0024]This invention is characterized by the 2nd information processing method comprising the following.

Regeneration steps in which encoded information in an AV stream reproduces start address information [ \*\*\*\* ] on the section, information which associates time information and address information in an AV stream, and time information of a characteristic picture in an AV stream as Clip information.

A control step which controls an output of an AV stream based on Clip information reproduced by processing of regeneration steps.

[0025] This invention is characterized by a program of the 2nd recording medium comprising the following.

Regeneration steps in which encoded information in an AV stream reproduces start address information [ \*\*\*\* ] on the section, information which associates time information and address information in an AV stream, and time information of a characteristic picture in an AV stream as Clip information.

A control step which controls an output of an AV stream based on Clip information reproduced by processing of regeneration steps.

[0026] The 2nd program of this invention Start address information [ \*\*\*\* / as Clip information / encoded information in an AV stream ] on the section, Regeneration steps which reproduce information which associates time information and address information in an AV stream, and time information of a characteristic picture in an AV stream, A program which performs a control step which controls an output of an AV stream based on Clip information reproduced by processing of regeneration steps.

[0027] As for a recording medium of this invention, start address information [ \*\*\*\* / encoded information in an AV stream ] on the section, information which associates time information and address information in an AV stream, and time information of a characteristic picture in an AV stream are recorded as Clip information.

[0028] In the 1st information processor of this invention and a method, a recording medium, and a program, As Clip information, start address information [ \*\*\*\* / encoded information in an AV stream ] on the section, information which associates time information and address information in an AV stream, and time information of a characteristic picture in an AV stream are recorded.

[0029] In the 2nd information processor of this invention and a method, a recording medium, and a program, As Clip information, start address information [ \*\*\*\* / encoded information in an AV stream ] on the section, information which associates time information and address information in an AV stream, and time information of a characteristic picture in an AV stream are reproduced.

[0030]

[Embodiment of the Invention] Below, an embodiment of the invention is described with reference to drawings. Drawing 1 is a figure showing the example of an internal configuration of the recording and reproducing device 1 which applied this invention.

First, the composition of the Records Department 2 which performs operation which records the signal inputted from the outside on a recording medium is explained. The recording and reproducing device 1 is considered as the composition which can input analog data or digital data and can be recorded.

[0031]The video signal of an analog is inputted into the terminal 11, and the audio signal of an analog is inputted into the terminal 12, respectively. The video signal inputted into the terminal 11 is outputted to the analyzing parts 14 and the AV encoder 15, respectively. The audio signal inputted into the terminal 12 is outputted to the analyzing parts 14 and the AV encoder 15. The analyzing parts 14 extract the focus, such as a scene change, from the inputted video signal and an audio signal.

[0032]The AV encoder 15 codes the video signal and audio signal which were inputted, respectively, and outputs system information (S), such as coding video stream (V), a coding audio stream (A), and AV synchronization, to the multiplexer 16.

[0033]A coding video stream is a video stream coded by MPEG(Moving Picture Expert Group) 2 method, for example, Coding audio streams are the audio stream coded by MPEG1 method, an audio stream coded by Dolby AC3 method (trademark), etc., for example. The multiplexer 16 multiplexes the stream of the inputted video and an audio based on input system information, and outputs it to the multiplexed stream analyzing parts 18 and saw spa KETTAIZA 19 via the switch 17.

[0034]Multiplexed streams are an MPEG2 transport stream and an MPEG 2 program stream, for example. Saw spa KETTAIZA 19 codes the inputted multiplexed stream to the AV stream which comprises a source packet according to the application format of the recording medium 100 on which the stream is made to record. Addition and the modulation process of an ECC code are performed by the ECC (error correction) coding part 20 and the modulation part 21, and an AV stream is outputted to the writing part 22. The writing part 22 writes an AV stream file in the recording medium 100 based on the control signal outputted from the control section 23 (it records).

[0035]Transport streams, such as digital television broadcasting inputted from a digital interface or a digital television tuner, are inputted into the terminal 13. They are a method which records those with two kind, and them on a transparent at the recording method of a transport stream inputted into the terminal 13, and a method recorded after carrying out re-encoding for the purposes, such as lowering a recording bit rate. The directions information on a recording method is inputted into the control section 23 from the terminal 24 as a user interface.

[0036]When recording an input transport stream on a transparent, the transport stream inputted into the terminal 13 is outputted to the multiplexed stream analyzing



parts 18 and saw spa KETTAIZA 19 via the switch 17. Since processing until an AV stream is recorded on the recording medium 100 after this is the same processing as the case where the above-mentioned input audio signal and video signal of an analog are coded and recorded, the explanation is omitted.

[0037]When recording after re-encoding an input transport stream, the transport stream inputted into the terminal 13 is inputted into the demultiplexer 26. The demultiplexer 26 performs demultiplex processing to the inputted transport stream, and extracts video stream (V), an audio stream (A), and system information (S).

[0038]A video stream is outputted to AV decoder 27 among the streams (information) extracted by the demultiplexer 26, and an audio stream and system information are outputted to the multiplexer 16, respectively. AV decoder 27 decodes the inputted video stream, and outputs the reproduced video signal to the AV encoder 15. The AV encoder 15 codes an input video signal, and outputs coding video stream (V) to the multiplexer 16.

[0039]The audio stream which was outputted from the demultiplexer 26 and inputted into the multiplexer 16 on the other hand, and system information, And the video stream outputted from the AV encoder 15 is multiplexed based on input system information, and is outputted to the multiplexed stream analyzing parts 18 and source packet TAIZA 19 via the switch 17 as a multiplexed stream. Since processing until an AV stream is recorded on the recording medium 100 after this is the same processing as the case where the above-mentioned input audio signal and video signal of an analog are coded and recorded, the explanation is omitted.

[0040]The recording and reproducing device 1 of this embodiment records the file of an AV stream on the recording medium 100, and it also records the application data base information explaining the file. Application data base information is created by the control section 23. The input to the control section 23 is the characteristic information of the video from the analyzing parts 14, the characteristic information of the AV stream from the multiplexed stream analyzing parts 18, and directions information from a user that it is inputted from the terminal 24.

[0041]The characteristic information of the video supplied from the analyzing parts 14 is generated by the analyzing parts 14 when the AV encoder 15 codes a video signal. The analyzing parts 14 analyze the contents of an input video signal and the audio signal, and generate the information related to the characteristic picture (clip mark) in an input dynamic image signal. This is the directions information on the picture of the characteristic points marking [ clip ], such as the starting point of the program in an input video signal, scene change point and start point [ of CM commercials ] – and a

point, a title, and a telop, for example, and the thumbnail of the picture is also contained in it. Furthermore, the switching point of the stereo of an audio signal and a monophonic recording and information, including a silent interval etc., are also included.

[0042]The directions information on these pictures is inputted into the multiplexer 16 via the control section 23. The multiplexer 16 returns the information for specifying the coding picture on an AV stream to the control section 23, when multiplexing the coding picture specified as a clip mark from the control section 23. Specifically, this information is the address information on PTS (presentation time stamp) of a picture, or the AV stream of that coding picture. The control section 23 associates and memorizes the information for specifying the kind and its coding picture of a characteristic picture on an AV stream.

[0043]The characteristic information of the AV stream from the multiplexed stream analyzing parts 18 is information related to the encoded information of the AV stream recorded, and is generated by the analyzing parts 18. For example, the change point information of the encoding parameter of the time stamp of I picture in an AV stream, address information and the discontinuous dot data of a system time clock, and an AV stream and the encoding parameter in an AV stream, etc. are included. When recording the transport stream inputted from the terminal 13 on a transparent, the multiplexed stream analyzing parts 18, The picture of the above-mentioned clip mark is detected out of an input transport stream, and the information for specifying the picture specified by the kind and clip mark is generated.

[0044]The directions information of the user from the terminal 24 is a bookmark, information on resume points, etc. which the character character and user explaining the specification information on the reproducing section specified by the user in an AV stream and the contents of the reproducing section set to a favorite scene.

[0045]Based on the above-mentioned input, the control section 23 The database of an AV stream (Clip), The management information (info.dvr) of the database of what (PlayList) carried out grouping of the reproducing section (PlayItem) of an AV stream, and the contents of record of the recording medium 100, and the information on a thumbnail image are created. Like an AV stream, the application data base information which comprises these information is processed by the ECC code-ized part 20 and the modulation part 21, and is inputted into the writing part 22. The writing part 22 records a database file on the recording medium 100 based on the control signal outputted from the control section 23.

[0046]The details about the application data base information mentioned above are

mentioned later.

[0047] Thus, the AV stream file (file of image data and voice data) recorded on the recording medium 100, When application data base information is reproduced by the regenerating section 3, the control section 23 directs to read application data base information from the recording medium 100 to the read section 28 first. And the read section 28 reads application data base information from the recording medium 100, and the application data base information is inputted into the control section 23 through a recovery and error correction processing of the demodulation section 29 and the ECC decoding part 30.

[0048] The control section 23 outputs the list of PlayList currently recorded on the recording medium 100 to the user interface of the terminal 24 based on application data base information. A user chooses PlayList to reproduce from the list of PlayList, and the information about PlayList which had reproduction specified is inputted into the control section 23. The control section 23 directs read-out of an AV stream file required for reproduction of the PlayList to the read section 28. The read section 28 reads an AV stream corresponding from the recording medium 100 according to the directions, and outputs it to the demodulation section 29. It gets over by performing predetermined processing, and the AV stream inputted into the demodulation section 29 is further outputted sauce DEPAKETTAIZA 31 through processing of the ECC decoding part 30.

[0049] Sauce DEPAKETTAIZA 31 is read from the recording medium 100, and changes the AV stream of an application format to which predetermined processing was performed into the stream which can process the demultiplexer 26. The demultiplexer 26 outputs system information (S), such as video stream (V) which constitutes the reproducing section (PlayItem) of an AV stream specified by the control section 23, an audio stream (A), and AV synchronization, to AV decoder 27. AV decoder 27 decodes a video stream and an audio stream, and outputs a reproduced video signal and a reproduced audio signal from the terminal 32 corresponding, respectively and the terminal 33.

[0050] When the information which directs random access reproduction and special reproduction is inputted from the terminal 24 as a user interface, the control section 23, Based on the contents of the database (Clip) of an AV stream, the reading position of the AV stream from the storage 100 is determined, and read-out of the AV stream is directed to the read section 28. For example, when reproducing PlayList with the selected user from predetermined time, the control section 23 directs to read the data with the time stamp nearest to the specified time from I picture to the read

section 28.

[0051] Out of the point of the program currently stored in ClipMark in Clip Information pulling out the head, or a scene change point. When a clip mark with a user is chosen (for example, this operation) The thumbnail picture list of the point of a program pulling out the head or scene change points currently stored in ClipMark is displayed on a user interface. Based on the contents of Clip Information, the control section 23 performed when a user chooses a certain picture from the inside determines the reading position of the AV stream from the recording medium 100, and directs read-out of the AV stream to the read section 28. That is, it directs to read the data from I picture in the address nearest to the address on the AV stream in which the picture which the user chose is stored to the read section 28. The read section 28 reads data from the specified address, and the read data, The AV information which is inputted into the demultiplexer 26, is decoded by AV decoder 27 through processing of the demodulation section 29, the ECC decoding part 30, and sauce DEPAKETTAIZA 31, and is shown in the address of the picture of a marking point is reproduced.

[0052] When fast reproduction (Fast-forward playback) is directed by the user, the control section 23, Based on the database (Clip) of an AV stream, it directs to read I-picture data in an AV stream continuously one by one to the read section 28.

[0053] The read section 28 reads the data of an AV stream from the specified random access point, and the read data is reproduced through processing of latter each part.

[0054] Next, a user explains the case where the AV stream currently recorded on the recording medium 100 is edited. When a user wants to specify the reproducing section of the AV stream currently recorded on the recording medium 100, and to create new salvage pathway, For example, from the popular music show of the program A, reproduce the singer's A portion and it continues after that, The information on the starting point (yne point) of a reproducing section and an end point (out point) is inputted into the control section 23 from the terminal 24 as a user interface to create the salvage pathway of liking to reproduce the portion of the singer A of the popular music show of the program B. The control section 23 creates the database of what (PlayList) carried out grouping of the reproducing section (PlayItem) of an AV stream.

[0055] When a user wants to eliminate a part of AV stream currently recorded on the recording medium 100, the information on the yne point of the elimination section and an out point is inputted into the control section 23 from the terminal 24 as a user interface. The control section 23 changes the database of PlayList so that only a required AV stream portion may be referred to. It directs to the writing part 22 so that

the unnecessary stream portion of an AV stream may be eliminated.

[0056]It is a case where a user wants to specify the reproducing section of the AV stream currently recorded on the recording medium 100, and to create new salvage pathway, and the case where he would like to connect each reproducing section seamlessly is explained. In such a case, the control section 23 creates the database of what (PlayList) carried out grouping of the reproducing section (PlayItem) of an AV stream, and performs partial re-encoding and re-multiplex-izing of a reproducing section of the video stream near a node further.

[0057]First, the information on the picture of the yne point of a reproducing section and the information on the picture of an out point are inputted into the control section 23 from the terminal 24. The control section 23 directs read-out of data required in order to reproduce the yne point side picture and the picture by the side of an out point to the read section 28. And the read section 28 reads data from the recording medium 100, and the data is outputted to the demultiplexer 26 through the demodulation section 29, the ECC decoding part 30, and sauce DEPAKETTAIZA 31.

[0058]The control section 23 analyzes the data inputted into the demultiplexer 26, A re multiplex-ized method is determined as the re-encoding method (change of picture\_coding\_type, assignment of the re-encoded encoding bit amount) of a video stream, and the method is supplied to the AV encoder 15 and the multiplexer 16.

[0059]Next, the demultiplexer 26 divides the inputted stream into video stream (V), an audio stream (A), and system information (S). A video stream has data inputted into AV decoder 27, and data inputted into the multiplexer 16. It is data required in order to re-encode the former data, and this is decoded by AV decoder 27, and the decoded picture is re-encoded with the AV encoder 15, and is made into a video stream. The latter data is data copied from an original stream without carrying out re-encoding. About an audio stream and system information, it is directly inputted into the multiplexer 16.

[0060]Based on the information inputted from the control section 23, the multiplexer 16 multiplexes an input stream and outputs a multiplexed stream. A multiplexed stream is processed by the ECC code-ized part 20 and the modulation part 21, and is inputted into the writing part 22. The writing part 22 records an AV stream on the recording medium 100 based on the control signal supplied from the control section 23.

[0061]Explanation about operation of the reproduction and edit based on application data base information and its information is given to below. Drawing 2 is a figure explaining the structure of an application format. An application format has two layers, PlayList and Clip, for management of an AV stream. Volume Information carries out

management of all the Clip(s) and PlayList(s) in a disk. Here, the pair of one AV stream and its attached information is considered to be one object, and it is called Clip. An AV stream file calls Clip AV stream file, and the attached information is called Clip Informationfile.

[0062]One Clip AV stream file stores the data which has arranged the MPEG2 transport stream in the structure in which it is specified by application format. Generally, although a file is treated as a sequence of bytes, the contents of Clip AV stream file are developed on a time-axis, and the entry point (I picture) in Clip is mainly specified in a hourly base. When the time stamp of the access point to predetermined Clip is given, Clip Information file is useful in order to find the address information which should start read-out of data in Clip AV stream file.

[0063]PlayList is explained with reference to drawing 3. PlayList chooses from Clip(s) the reproducing section which a user wants to see, and it is provided in order to be able to edit it easily. One PlayList is a meeting of the reproducing section in Clip. One reproducing section in predetermined Clip is called PlayItem, and it is expressed with the pair of the yne point (IN) on a time-axis, and an out point (OUT). Therefore, PlayList is constituted when two or more PlayItem(s) gather.

[0064]There are two types of PlayList(s). One is Real PlayList and another is Virtual PlayList. Real PlayList is sharing the stream portion of Clip which it is referring to. That is, when Real PlayList occupies in a disk the data volume equivalent to the stream portion of Clip which is referring to it and Real PlayList is eliminated, data is eliminated also for the stream portion of Clip which it is referring to.

[0065]Virtual PlayList is not sharing the data of Clip. Therefore, even if Virtual PlayList is changed or eliminated, by the contents of Clip, change does not arise at all.

[0066]Next, edit of Real PlayList is explained. Drawing 4 (A) is a figure about the creation (create: creation) of Real PlayList, and when an AV stream is recorded as new Clip, it is operation in which RealPlayList which refers to the whole Clip is newly created.

[0067]Drawing 4 (B) is a figure about the divide (divide: division) of Real PlayList, and is operation in which Real PlayList is divided at a point [ \*\*\*\* ] and divided into two Real PlayList. For example in one clip managed by one PlayList, when two programs are managed, in registration (record), a user does the operation of this division again as each program, and it is performed at the time of being \*\*\*\*\*. There is nothing for which the contents of Clip are changed by this operation (the Clip itself is divided).

[0068]Drawing 4 (C) is a figure about the combine (combine: combination) of Real

PlayList, and is operation which combines two Real PlayList and is set to one new Real PlayList. A user reregisters two programs as one program, and operation of this combination is performed at the time of being \*\*\*\*\*, for example. There is nothing for which Clip is changed by this operation (the Clip itself is set to one).

[0069]Drawing 5 (A) is a figure about deletion (delete: deletion) of whole Real PlayList, When operation which eliminates whole predetermined Real PlayList is carried out, the stream portion to which Clip which deleted Real PlayList refers to corresponds is also deleted.

[0070]Drawing 5 (B) is a figure about partial deletion of Real PlayList, and when a portion [ \*\*\*\* / Real PlayList ] is deleted, it is changed so that corresponding PlayItem may refer to only the stream portion of required Clip. And the stream portion to which Clip corresponds is deleted.

[0071]Drawing 5 (C) is a figure about minimization (Minimize: minimization) of Real PlayList, It is operation of referring to only the stream portion of Clip required for Virtual PlayList for PlayItem corresponding to Real PlayList. The stream portion to which Clip unnecessary for Virtual PlayList corresponds is deleted.

[0072]Real PlayList is changed by the operation which was mentioned above, When the stream portion of Clip which the Real PlayList refers to is deleted, Virtual PlayList which is using the deleted Clip may exist, and a problem may arise by deleted Clip in the Virtual PlayList.

[0073]As opposed to operation of [ so that such a thing may not arise ] deletion to a user, "If Virtual PlayList which is referring to the stream portion of Clip which the Real PlayList is referring to exists and the Real PlayList is eliminated, although the Virtual PlayList will also be eliminated, is it still good? processing of the deletion with a user's directions after urging a check (warning) by displaying the message " etc. -- execution -- or it cancels. Or operation of minimization is made to be performed instead of deleting Virtual PlayList to Real PlayList.

[0074]Next, the operation to Virtual PlayList is explained. The contents of Clip are not changed even if operation is performed to Virtual PlayList. Drawing 6 is assembling (Assemble). Edit (IN-OUT edit) It is a related figure and is operation of making PlayItem of the reproducing section for which it asked when the user wanted to see, and creating Virtual PlayList. The seamless connection between PlayItem(s) is supported by the application format (after-mentioned).

[0075]As shown in drawing 6 (A), two Real PlayList1 and 2, When Clip1 corresponding to each RealPlayList and 2 exist, A user points to the predetermined section

(section-layItem1 to In1 thru/or Out1) in Real PlayList1 as a reproducing section, and as the section reproduced continuously, When it points to the predetermined section (section-layItem2 to In2 thru/or Out2) in Real PlayList2 as a reproducing section, As shown in drawing 6 (B), one Virtual PlayList which comprises PlayItem1 and PlayItem2 is created.

[0076]Next, the reorganization collection (Re-editing) of Virtual PlayList is explained. In a reorganization collection, change of the yne point in Virtual PlayList, and an out point, There are insertion (insert) of new PlayItem to Virtual PlayList, an addition (append), deletion of PlayItem in Virtual PlayList, etc. Virtual PlayList itself can also be deleted.

[0077]Drawing 7 is a figure about postrecording (Audio dubbing (post recording)) of the audio to Virtual PlayList, and is operation which registers postrecording of the audio to VirtualPlayList as a sub path. Postrecording of this audio is supported by the application format. An additional audio stream is added to the AV stream of the main path of Virtual PlayList as a sub path.

[0078]As operation common to Real PlayList and Virtual PlayList, there is change (Moving) of the reproduction sequence of PlayList as shown in drawing 8. This operation is change of the reproduction sequence of PlayList in the inside of a disk (volume), and is supported by Table Of PlayList (with reference to drawing 20 etc., it mentions later) defined in an application format. The contents of Clip are not changed by this operation.

[0079]Next, the mark (Mark) is explained. The mark is provided in order to specify the highlight in Clip and PlayList, and characteristic time, as shown in drawing 9. The mark added to Clip is called ClipMark (clip mark). Specify the characteristic scene resulting from the contents of the AV stream, for example, ClipMark is a head broth point, a scene change point, etc. of a program. ClipMark is generated by the analyzing parts 14 of drawing 1. When reproducing PlayList, it can be used with reference to the mark of Clip which the PlayList refers to.

[0080]The mark added to PlayList is called PlayListMark (play list mark). Are mainly set by the user, for example, PlayListMark(s) are a bookmark, resume points, etc. Setting a mark to Clip or PlayList is performed by adding the time stamp in which the time of a mark is shown to a mark list. Deleting a mark is removing the time stamp of the mark out of a mark list. Therefore, as for an AV stream, a change of what is not made by setting out or deletion of a mark, either.

[0081]It may be made to specify the picture which ClipMark refers to on an address basis in the inside of an AV stream as another format of ClipMark. Setting a mark to



Clip is performed by adding the information on an address base which shows the picture of a marking point to a mark list. Deleting a mark is removing the information on an address base which shows the picture of the marking point out of a mark list. Therefore, as for an AV stream, a change of what is not made by setting out or deletion of a mark, either.

[0082]Next, a thumbnail is explained. A thumbnail is a still picture added to Volume, Playlist, and Clip. There are two kinds of thumbnails and one is a thumbnail as representation drawing showing the contents. This is used by the menu screen for choosing the thing a user mainly wants to operate and look at cursor (un-illustrating) etc. Another is a picture showing the scene which the mark has pointed out.

[0083]Volume and each Playlist need to enable it to have representation drawing. The representation drawing of Volume is a disk (the recording medium 100 presupposes that it is a disk-like thing, and the recording medium 100 and the following). suitably -- a disk -- describing -- when it sets to the predetermined place of the recording and reproducing device 1, it assumes being used when displaying the still picture showing the contents of the disk first. In the menu screen which chooses Playlist, the representation drawing of Playlist assumes being used as a still picture for expressing the contents of Playlist.

[0084]Although it is possible as representation drawing of Playlist to make the picture of the beginning of Playlist into a thumbnail (representation drawing), when the picture of the head of the regeneration time 0 expresses the contents, it is not necessarily the optimal picture. Then, a user enables it to set up arbitrary pictures as a thumbnail of Playlist. Two kinds of thumbnails, the thumbnail as representation drawing which expresses Volume above, and the thumbnail as representation drawing showing Playlist, are called a menu thumbnail. Since a menu thumbnail is displayed frequently, it needs to be read from a disk at high speed. For this reason, it is efficient to store all the menu thumbnails in one file. It is not necessary to be necessarily the picture extracted from the animation in volume, and as shown in drawing 10, a menu thumbnail may be taken from a personal computer or a digital still camera, and a \*\*\*\*\* picture may be sufficient as it.

[0085]It can be necessary to strike two or more marks, and in order to know the contents of the mark position, it is necessary to enable it to see the picture of a marking point easily to Clip and Playlist on the other hand. The picture showing such a marking point is called the mark thumbnail (Mark Thumbnails). Therefore, the picture which becomes the origin of a mark thumbnail becomes more nearly main [ what extracted the picture of the marking point ] than the picture captured from the

exterior.

[0086]Drawing 11 is a mark attached to PlayList, and a figure showing the relation of the mark thumbnail, and drawing 12 is a mark attached to Clip, and a figure showing the relation of the mark thumbnail. Since a mark thumbnail is used with a sub menu etc. when the details of Playlist are expressed unlike a menu thumbnail, what it is read in short access time is not required. Therefore, whenever a thumbnail is needed, the recording and reproducing device 1 opens a file, and it does not become a problem even if it takes time somewhat by reading a part of the file.

[0087]In order to reduce the number of files which exists in volume, all the mark thumbnails are good to store in one file. Although Playlist can have one menu thumbnail and two or more mark thumbnails, since Clip does not have the necessity that a direct user chooses (it usually specifies via Playlist), it does not need to provide a menu thumbnail.

[0088]Drawing 13 is a figure showing the relation of the menu thumbnail at the time of taking having mentioned above into consideration, a mark thumbnail, PlayList, and Clip. The menu thumbnail provided in the menu thumbnail file for every PlayList is filed. The volume thumbnail representing the contents of the data currently recorded on the disk is contained in the menu thumbnail file. The thumbnail by which the mark thumbnail file was created for every PlayList and every Clip is filed.

[0089]Next, CPI (Characteristic Point Information) is explained. CPI is data contained in Clip information files, When the time stamp of the access point to Clip is given, it is mainly used in order to find the data address which should start read-out of data in Clip AV stream file. According to this embodiment, two kinds of CPI(s) are used. One is EP\_map and another is TU\_map.

[0090]EP\_map is a list of entry point (EP) data, and it is extracted from an elementary stream and a transport stream. This has the address information for finding the place of the entry point which should start decoding in an AV stream. One EP data comprises a pair of the data address in the AV stream of a presentation time stamp (PTS) and the access unit corresponding to the PTS.

[0091]EP\_map is mainly used for two purposes. It is used in order to find the data address in the AV stream of the access unit referred to [ 1st ] with a presentation time stamp in PlayList. It is used for the 2nd for first forward reproduction or first reverse reproduction. When the recording and reproducing device 1 records an input AV stream and the syntax of the stream can be analyzed, EP\_map is created and it is recorded on a disk.

[0092]TU\_map has a list of the time unit (TU) data based on the arrival time of the

transport packet inputted through a digital interface. This gives the relation between the time of an arrival time base, and the data address in an AV stream. When the recording and reproducing device 1 records an input AV stream and the syntax of the stream cannot be analyzed, TU\_map is created and it is recorded on a disk.

[0093]STCInfo stores the discontinuous dot data of STC in the AV stream file which is storing the MPEG2 transport stream.

[0094]When an AV stream has a break point of STC, PTS of the same value may appear in the AV stream file. Therefore, when pointing out a certain time on an AV stream on a PTS basis, just PTS of an access point is insufficient in order to specify the point. The index of the STC section [ \*\*\*\* ] containing the PTS is required. The STC section [ \*\*\*\* ] is called STC-sequence in this format, and that index is called STC-sequence-id. The information on STC-sequence is defined by STCInfo of Clip Information file.

[0095]STC-sequence-id is an option in the AV stream file which uses it by an AV stream file with EP\_map, and has TU\_map.

[0096]A program is a meeting of elemental stream and shares only one system time base for the synchronous reproduction of these streams.

[0097]What the contents of the AV stream understand in advance of decoding of an AV stream for playback equipment (recording and reproducing device 1 of drawing 1) is useful. For example, they are information, including the value of PID of the transport packet which transmits the elementary stream of video or an audio, video, the component kind of audio, etc., (for example, the videos of HDTV, the audio streams of MPEG-2 AAC, etc.). This information is useful although the menu screen which explains to a user the contents of PlayList which refers to an AV stream is created, and it is useful in order to set the AV decoder of playback equipment, and the initial state of a demultiplexer in advance of decoding of an AV stream.

[0098]For this reason, Clip Information file has ProgramInfo for explaining the contents of the program.

[0099]In the AV stream file which is storing the MPEG2 transport stream, the contents of a program may change in a file. For example, it is that PID of the transport packet which transmits a video elementary stream changes, or the component kind of video stream changes from SDTV to HDTV etc.

[0100]ProgramInfo stores the information on the change point of the contents of a program in the inside of an AV stream file. In an AV stream file, the contents of a program defined in this format call the fixed section Program-sequence.

[0101]Program-sequence is an option in the AV stream file which uses it by an AV

stream file with EP\_map, and has TU\_map.

[0102]This embodiment defines the stream format (SESF) of self encoding. SESF is used when coding to an MPEG2 transport stream, after decoding the purpose of coding an analog input signal, and a digital input signal (for example, DV).

[0103]SESF defines coding restrictions of the elementary stream about MPEG-2 transport stream and an AV stream. When the recording and reproducing device 1 encodes and records a SESF stream, EP\_map is created and it is recorded on a disk.

[0104]Either of the methods shown below is used and the stream of digital broadcasting is recorded on the recording medium 100. First, transformer coding of the stream of digital broadcasting is carried out at a SESF stream. In this case, the recorded stream must be based on SESF. In this case, EP\_map must be created and it must be recorded on a disk.

[0105]Or transformer coding is carried out at new elemental stream, and the elementary stream which constitutes a digital broadcasting stream is re-multiplexed to the new transport stream based on the stream format which the standardization organization of the digital broadcasting stream defines. In this case, EP\_map must be created and it must be recorded on a disk.

[0106]For example, an input stream is MPEG-2 transport stream of ISDB (standard name of digital BS broadcasting of Japan) conformity, and suppose that it contains a HDTV video stream and a MPEG AAC audio stream. Transformer coding of the HDTV video stream is carried out at a SDTV video stream, and the SDTV video stream and an original AAC audio stream are re-multiplexed to TS. Both the transport streams recorded as a SDTV stream must be based on an ISDB format.

[0107]The stream of digital broadcasting as other methods at the time of being recorded on the recording medium 100, It is a case (it records without changing any input transport streams) where an input transport stream is recorded on a transparent, and EP\_map is then created and it is recorded on a disk.

[0108]Or it is a case (it records without changing any input transport streams) where an input transport stream is recorded on a transparent, and TU\_map is then created and it is recorded on a disk.

[0109]Next, a directory and a file are explained. Hereafter, the recording and reproducing device 1 is suitably described to be DVR (Digital Video Recording).

Drawing 14 is a figure showing an example of the directory structure on a disk. A directory required on the disk of DVR, As shown in drawing 14, they are a root directory including a "DVR" directory, a "PLAYLIST" directory, a "CLIPINF" directory, and a "DVR" directory including an "M2TS" directory and "DATA"

directory. Although directories other than these may be made to be created under a root directory, they presuppose that it is ignored in the application format of this embodiment.

[0110]All the files and directories which are specified by DVR application format under a "DVR" directory are stored. A "DVR" directory includes four directories. Under a "PLAYLIST" directory, the database file of Real PlayList and Virtual PlayList is placed. This directory exists, even if one does not have PlayList.

[0111]The database of Clip is placed under a "CLIPINF" directory. This directory also exists, even if one does not have Clip. An AV stream file is placed under an "M2TS" directory. This directory exists, even if one does not have an AV stream file. In the "DATA" directory, the file of data broadcasting, such as digital TV broadcasting, is stored.

[0112]A "DVR" directory stores the file shown below. It is made under a "info.dvr" file and a DVR directory, and the overall information on an application layer is stored. Only one info.dvr must be under a DVR directory. A file name presupposes that it is fixed to info.dvr. A "menu.thmb" file stores the information relevant to a menu thumbnail image. Zero or one menu thumbnail must be under a DVR directory. A file name presupposes that it is fixed to memu.thmb. When one does not have a menu thumbnail image, this file does not need to exist.

[0113]A "mark.thmb" file stores the information relevant to a mark thumbnail image. Zero or one mark thumbnail must be under a DVR directory. A file name presupposes that it is fixed to mark.thmb. When one does not have a menu thumbnail image, this file does not need to exist.

[0114]A "PLAYLIST" directory stores two kinds of PlayList files, and they are Real PlayList and Virtual PlayList. "xxxxx.rpls" A file stores the information relevant to one Real PlayList. One file is made for every Real PlayList. A file name is "xxxxx.rpls." Here, "xxxxx" is a number to five 0 thru/or 9. A file extension child presupposes that it must be "rpls".

[0115]A "yyyyy.vpls" file stores the information relevant to one Virtual PlayList. One file is made for every Virtual PlayList. A file name is "yyyyy.vpls." Here, "yyyyy" is a number to five 0 thru/or 9. A file extension child presupposes that it must be "vpls".

[0116]A "CLIPINF" directory stores one file corresponding to each AV stream file. "zzzzz.clpi" A file is Clip Information file corresponding to one AV stream file (Clip AV stream file or Bridge-Clip AV stream file). A file name is "zzzzz.clpi" and "zzzzz" is a number to five 0 thru/or 9. A file extension child presupposes that it must be "clpi".

[0117]An "M2TS" directory stores the file of an AV stream. A "zzzzz.m2ts" file is an

AV stream file treated by a DVR system. This is Clip AV stream file or Bridge-Clip AV stream. A file name is "zzzzz.m2ts" and "zzzzz" is a number to five 0 thru/or 9. A file extension child presupposes that it must be "m2ts."

[0118]The "DATA" directory stores the data transmitted from data broadcasting, and data is XML file, an MHEG file, etc., for example.

[0119]Next, the syntax and semantics of each directory (file) are explained. First, a "info.dvr" file is explained. Drawing 15 is a figure showing the syntax of a "info.dvr" file. A "info.dvr" file comprises three objects and they are DVRVolume(), TableOfPlayLists(), and MakersPrivateData().

[0120]To explain the syntax of info.dvr shown in drawing 15

TableOfPlayLists\_Start\_address, The start address of TableOfPlayList() is shown by making the relative number of bytes from the byte of the head of an info.dvr file into a unit. A relative number of bytes is counted from zero.

[0121]MakersPrivateData\_Start\_address shows the start address of MakersPrivateData() by making the relative number of bytes from the byte of the head of an info.dvr file into a unit. A relative number of bytes is counted from zero. padding\_word (padding word) is inserted according to the syntax of info.dvr. N1 and N2 are zero or arbitrary positive integers. It may be made for each padding word to take any value.

[0122]DVRVolume() stores the information which describes the contents of volume (disk). Drawing 16 is a figure showing the syntax of DVRVolume(). version\_number shows four character characters which show the version number of this DVRVolume() for explaining the syntax of DVR Volume() shown in drawing 16. version\_number is coded with "0045" according to ISO 646.

[0123]length is expressed with the 32-bit unsigned integer which shows the number of bytes of DVRVolume() from immediately after this length field to the last of DVRVolume().

[0124]ResumeVolume() has memorized the file name of Real PlayList reproduced at the end in volume, or Virtual PlayList. However, when a user interrupts reproduction of Real PlayList or Virtual PlayList, a playback position is stored in resume-mark defined in PlayListMark() (drawing 42, drawing 43).

[0125]Drawing 17 is a figure showing the syntax of ResumeVolume(). The syntax of ResumeVolume() shown in drawing 17 to explain valid\_flag, When it is shown that the resume\_PlayList\_name field is effective when this 1-bit flag is set to 1 and this flag is set to 0, it is shown that the resume\_PlayList\_name field is invalid.

[0126]10 bytes of field of resume\_PlayList\_name shows the file name of Real PlayList

by which resume should be carried out, or Virtual PlayList.

[0127]UIAppInfoVolume in the syntax of DVRVolume() shown in drawing 16 stores the parameter of the user interface application about volume. Drawing 18 is a figure showing the syntax of UIAppInfoVolume, and the 8-bit field of character\_set shows the encoding method of the character character coded in the Volume\_name field for explaining the semantics. The encoding method corresponds to the value shown in drawing 19.

[0128]Eight bit fields of name\_length show the byte length of the volume name shown in the Volume\_name field. The field of Volume\_name shows the name of volume. The number of bytes of the left in this field to a name\_length number is an effective character character, and it shows the name of volume. In the Volume\_name field, what kind of value may be [ value after these effective character character ] contained.

[0129]Volume\_protect\_flag is a flag which shows whether the contents in volume may be shown without restricting to a user. Only when this flag is set to 1 and a user is able to input a PIN number (password) correctly, showing a user the contents of that volume (reproduced) is permitted. When this flag is set to 0, even if a user does not input a PIN number, showing a user the contents of that volume is permitted.

[0130]First, when a user inserts a disk in a player. [ whether this flag is set to 0, and ] Or if a user is able to input a PIN number correctly even if this flag is set to 1, the recording and reproducing device 1 will display the list of PlayList in that disk. Reproduction restrictions of each PlayList are unrelated to volume\_protect\_flag, and it is shown by playback\_control\_flag defined in UIAppInfoPlayList().

[0131]PIN comprises a number to four 0 thru/or 9, and each number is coded according to ISO/IEC646. The field of ref\_thumbnail\_index shows the information on the thumbnail image added to volume. In the case of the value whose ref\_thumbnail\_index field is not 0xFFFF, the thumbnail image is added to the volume and the thumbnail image is stored in a menu.thum file. The picture is referred to using the value of ref\_thumbnail\_index in a menu.thum file. When the ref\_thumbnail\_index field is 0xFFFF, it is shown that the thumbnail image is not added to the volume.

[0132]Next, TableOfPlayLists() in the syntax of info.dvr shown in drawing 15 is explained. TableOfPlayLists() stores the file name of PlayList (Real PlayList and Virtual PlayList). All the PlayList files currently recorded on volume are included in TableOfPlayList(). TableOfPlayLists() shows the default reproduction sequence of PlayList in volume.

[0133]Drawing 20 is a figure showing the syntax of TableOfPlayLists(), version\_number of TableOfPlayLists shows four character characters which show the version number

of this TableOfPlayLists for explaining that syntax. version\_number must be coded with "0045" according to ISO 646.

[0134]length is an integer without 32-bit numerals which shows the number of bytes of TableOfPlayLists() from immediately after this length field to the last of TableOfPlayLists(). The 16-bit field of number\_of\_PlayLists shows the loop count of for-loop containing Playlist\_file\_name. This number must be equal to the number of Playlist(s) currently recorded on volume. 10 bytes of number of Playlist\_file\_name shows the file name of Playlist.

[0135]Drawing 21 is a figure showing another composition of the syntax of TableOfPlayLists(). The syntax shown in drawing 21 is considered as the composition which included UIAppinfoPlaylist (after-mentioned) in the syntax shown in drawing 20. Thus, it becomes possible only by reading TableOfPlayLists to create a menu screen by having composition in which UIAppinfoPlaylist was included. Here, the following explanation is given noting that the syntax shown in drawing 20 is used.

[0136]MakersPrivateData in the syntax of info.dvr shown in drawing 15 is explained. MakersPrivateData is provided so that the maker of the recording and reproducing device 1 can insert the private data of a maker into MakersPrivateData() for the special application of each company. The private data of each maker has maker\_ID standardized in order to identify the maker which defined it. MakersPrivateData() may also contain one or more maker\_ID.

[0137]When a predetermined maker wants to insert private data and the private data of other makers is already contained in MakersPrivateData(), other makers, The old private data which already exists is not eliminated, but new private data is added into MakersPrivateData(). Thus, in this embodiment, the private data of two or more makers carries out as [ be / being contained in one MakersPrivateData() / possible ].

[0138]Drawing 22 is a figure showing the syntax of MakersPrivateData.

version\_number shows four character characters which show the version number of this MakersPrivateData() for explaining the syntax of MakersPrivateData shown in drawing 22. version\_number must be coded with "0045" according to ISO 646. length shows the 32-bit unsigned integer which shows the number of bytes of MakersPrivateData() from immediately after this length field to the last of MakersPrivateData().

[0139]mpd\_blocks\_start\_address shows the head byte address of the first mpd\_block() by making the relative number of bytes from the byte of the head of MakersPrivateData() into a unit. A relative number of bytes is counted from zero. number\_of\_maker\_entries is a 16-bit unsigned integer which gives the number of



entries of the maker private data contained in `MakersPrivateData()`. Two or more maker private data which have a value of the same `maker_ID` in `MakersPrivateData()` must not exist.

[0140]`mpd_block_size` is a 16-bit unsigned integer which gives the size of one `mpd_block` by making 1024 bytes into a unit. For example, if it becomes `mpd_block_size=1`, it shows that the size of one `mpd_block` is 1024 bytes.

`number_of_mpd_blocks` is a 16-bit unsigned integer which gives the number of `mpd_block` contained in `MakersPrivateData()`. `maker_ID` is a 16-bit unsigned integer which shows the manufacturing maker of the DVR system which created the maker private data. The value coded by `maker_ID` is specified by the licenser of this DVR format.

[0141]`maker_model_code` is a 16-bit unsigned integer which shows the model number code of the DVR system which created the maker private data. The value coded by `maker_model_code` is set up by the manufacturing maker who received the license of this format. `start_mpd_block_number` is a 16-bit unsigned integer which shows the number of `mpd_block` by which the maker private data is started. The aryne of the initial data of maker private data must be carried out to the head of `mpd_block`. `start_mpd_block_number` corresponds to the variable `j` in for-loop of `mpd_block`.

[0142]`mpd_length` is a 32-bit unsigned integer which shows the size of maker private data per byte. `mpd_block` is a field in which maker private data is stored. All the `mpd_block` in `MakersPrivateData()` must be the same sizes.

[0143]Next, `xxxxx.rpls` and `yyyyy.vpls` will be explained if it puts in another way about Real PlayList file and Virtual PlayList file. Drawing 23 is a figure showing the syntax of `xxxxx.rpls` (Real PlayList) or `yyyyy.vpls` (Virtual PlayList). `xxxxx.rpls` and `yyyyy.vpls` have the same syntax composition. `xxxxx.rpls` and `yyyyy.vpls` comprise three objects, respectively and they are `PlayList()`, `PlayListMark()`, and `MakersPrivateData()`.

[0144]`PlayListMark_Start_address` shows the start address of `PlayListMark()` by making the relative number of bytes from the byte of the head of a PlayList file into a unit. A relative number of bytes is counted from zero.

[0145]`MakersPrivateData_Start_address` shows the start address of `MakersPrivateData()` by making the relative number of bytes from the byte of the head of a PlayList file into a unit. A relative number of bytes is counted from zero.

[0146]`padding_word` (padding word) is inserted according to the syntax of a PlayList file, and `N1` and `N2` are zero or arbitrary positive integers. It may be made for each padding word to take any value.

[0147]Here, although already explained simple, PlayList is explained further. Refer to the reproducing section in all the Clip(s) except Bridge-Clip (after-mentioned) for all the Real PlayList in a disk. And two or more RealPlayList(s) must not make the reproducing section shown by those PlayItem(s) overlap in the same Clip.

[0148]With reference to drawing 24, as shown, Real PlayList to which all the Clip(s) correspond exists in explaining further at drawing 24 (A). This rule is followed after editing work is performed, as shown in drawing 24 (B). therefore, all the Clip(s) -- which -- it is -- certainly viewing and listening is possible by referring to Real PlayList.

[0149]As shown in drawing 24 (C), the reproducing section of Virtual PlayList must be included in the reproducing section of Real PlayList, or the reproducing section of Bridge-Clip. Bridge-Clip referred to at no Virtual PlayList must not exist in a disk.

[0150]Although Real PlayList includes the list of PlayItem, it must not contain SubPlayItem. Virtual PlayList includes the list of PlayItem, CPI\_type shown in PlayList() is EP\_map type, And when PlayList\_type is 0 (PlayList containing video and an audio), Virtual PlayList can contain one SubPlayItem. In PlayList() in this embodiment, SubPlayItem must be used only for the purpose of postrecording of an audio and the number of SubPlayItem(s) which one Virtual PlayList has must be 0 or 1.

[0151]Next, PlayList is explained. Drawing 25 is a figure showing the syntax of PlayList. They are four character characters in which version\_number shows the version number of this PlayList() for explaining the syntax of PlayList shown in drawing 25. version\_number must be coded with "0045" according to ISO 646. length is a 32-bit unsigned integer which shows the number of bytes of PlayList() from immediately after this length field to the last of PlayList(). PlayList\_type is the 8-bit field which shows the type of this PlayList, and shows drawing 26 that example.

[0152]CPI\_type is a 1-bit flag and shows the value of CPI\_type of Clip referred to by PlayItem() and SubPlayItem(). If all Clip(s) referred to by one PlayList do not have a the same value of CPI\_type defined in those CPI(), they will not become. number\_of\_PlayItems is the 16-bit field which shows the number of PlayItem(s) in PlayList.

[0153]PlayItem\_id corresponding to predetermined PlayItem() is defined by the turn that the PlayItem() appears, in for-loop containing PlayItem(). PlayItem\_id is started from 0. number\_of\_SubPlayItems is the 16-bit field which shows the number of SubPlayItem(s) in PlayList. This value is 0 or 1. The path (audio stream path) of an additional audio stream is a kind of a sub path.

[0154]Next, UIAppInfoPlayList of the syntax of PlayList shown in drawing 25 is explained. UIAppInfoPlayList stores the parameter of the user interface application

about PlayList. Drawing 27 is a figure showing the syntax of UIAppInfoPlayList. For explaining the syntax of UIAppInfoPlayList shown in drawing 27, character\_set is the 8-bit field and shows the encoding method of the character coded in the PlayList\_name field. The encoding method corresponds to the value based on the table shown in drawing 19.

[0155]name\_length is eight bit fields and shows the byte length of the PlayList name shown in the PlayList\_name field. The field of PlayList\_name shows the name of PlayList. The number of bytes of the left in this field to a name\_length number is an effective character character, and it shows the name of PlayList. In the PlayList\_name field, what kind of value may be [ value after these effective character character ] contained.

[0156]record\_time\_and\_date is the 56-bit field in which time when PlayList is recorded is stored. This field codes 14 numbers by 4-bit Binary Coded Decimal (BCD) about a /part / second at the time of year / moon / day/. For example, 2001/12/23:01:02:03 are coded with "0x20011223010203."

[0157]duration is the 24-bit field which showed the total reproduction time of PlayList in the unit of time / part / second. This field codes six numbers by 4-bit Binary CodedDecimal (BCD). For example, 01:45:30 is coded with "0x014530."

[0158]valid\_period is the 32-bit field which shows the period when PlayList is effective. This field codes eight numbers by 4-bit Binary Coded Decimal (BCD). For example, the recording and reproducing device 1 is used as it said that automatic deletion of the PlayList over which this shelf-life passed was carried out. For example, 2001/05/07 are coded with "0x20010507."

[0159]maker\_id is a 16-bit unsigned integer which shows the manufacturer of the DVR player (recording and reproducing device 1) which updated the PlayList at the end. The value coded by maker\_id is assigned by the licenser of a DVR format. maker\_code is a 16-bit unsigned integer which shows the model number of the DVR player which updated the PlayList at the end. The value coded by maker\_code is decided by the manufacturer who received the license of the DVR format.

[0160]The PlayList is reproduced, only when the flag of playback\_control\_flag is set to 1 and a user is able to input a PIN number correctly. When this flag is set to 0, even if a user does not input a PIN number, the user can view and listen to that PlayList.

[0161]As a table is shown write\_protect\_flag in drawing 28 (A), when being set to 1, write\_protect\_flag is removed, and the contents of the PlayList are not eliminated and changed. When this flag is set to 0, the user can eliminate and change that PlayList freely. When this flag is set to 1, before a user eliminates, edits or overwrites that

PlayList, the recording and reproducing device 1 displays a message which is reconfirmed to a user.

[0162]Real PlayList by which write\_protect\_flag is set to zero exists, And Virtual PlayList which refers to Clip of the Real PlayList exists, and write\_protect\_flag of the Virtual PlayList may be set to one. When a user is going to eliminate RealPlayList, the recording and reproducing device 1, "Minimize" [ it warns a user of existence of above-mentioned Virtual PlayList, or / the Real PlayList ] before eliminating the Real PlayList.

[0163]As shown in drawing 28 (B), when the flag is set to 1, is\_played\_flag the PlayList, After being recorded, when having been reproduced is shown and it is set to 0 at once, the PlayList shows not being reproduced once, after being recorded.

[0164]archive is the 2-bit field which shows whether the PlayList is original or it is copied, as shown in drawing 28 (C). The field of ref\_thumbnail\_index shows the information on a thumbnail image that PlayList is represented. In the case of the value whose ref\_thumbnail\_index field is not 0xFFFF, the thumbnail image representing PlayList is added to the PlayList, and the thumbnail image is stored in a menu.thum file. The picture is referred to using the value of ref\_thumbnail\_index in a menu.thum file. When the ref\_thumbnail\_index field is 0xFFFF, the thumbnail image representing PlayList is not added to the PlayList.

[0165]Next, PlayItem is explained. One PlayItem() contains the following data fundamentally. Clip\_information\_file\_name for specifying the file name of Clip, The pair of IN\_time for pinpointing the reproducing section of Clip, and OUT\_time, When CPI\_type defined in PlayList() is EP\_map type, They are STC\_sequence\_id which IN\_time and OUT\_time refer to, and connection\_condition which shows the state of connection between PlayItem to precede and the present PlayItem.

[0166]Those PlayItem(s) are arranged in on the global time-axis of PlayList without the gap of time, or overlap by the single tier when PlayList comprises two or more PlayItem(s). CPI\_type defined in PlayList() is EP\_map type, And the pair of IN\_time defined in the PlayItem when the present PlayItem does not have BridgeSequence(), and OUT\_time, The time on the same STC continuation section specified by STC\_sequence\_id must be pointed out. Such an example is shown in drawing 29.

[0167]CPI\_type defined in PlayList() is EP\_map type, and drawing 30 shows the case where the rule explained below is applied, when the present PlayItem has BridgeSequence(). IN\_time (what is indicated to be IN\_time1 in the figure) of PlayItem preceded with the present PlayItem has pointed out the time on the STC continuation section specified by STC\_sequence\_id of PlayItem to precede. OUT\_time (what is

indicated to be OUT\_time1 in the figure) of PlayItem to precede has pointed out the time in Bridge-Clip specified in BridgeSequenceInfo() of the present PlayItem. This OUT\_time must follow the coding restrictions mentioned later.

[0168]IN\_time (what is indicated to be IN\_time2 in the figure) of the present PlayItem has pointed out the time in Bridge-Clip specified in BridgeSequenceInfo() of the present PlayItem. This IN\_time must also follow the coding restrictions mentioned later. OUT\_time (what is indicated to be OUT\_time2 in the figure) of PlayItem of the present PlayItem has pointed out the time on the STC continuation section specified by STC\_sequence\_id of the present PlayItem.

[0169]As shown in drawing 31, when CPI\_type of PlayList() is TU\_map type, the pair of IN\_time of PlayItem and OUT\_time has pointed out the time on the same Clip AV stream.

[0170]The syntax of PlayItem comes to be shown in drawing 32. The field of Clip\_Information\_file\_name shows the file name of ClipInformation file for explaining the syntax of PlayItem shown in drawing 32. Clip\_stream\_type defined in ClipInfo() of this Clip Information file must show Clip AV stream.

[0171]STC\_sequence\_id is the 8-bit field and shows STC\_sequence\_id of the STC continuation section which PlayItem refers to. When CPI\_type specified in PlayList() is TU\_map type, these eight bit fields have no meanings, but are set to 0. IN\_time is 32 bit fields and stores the reproduction start time of PlayItem. The semantics of IN\_time changes with CPI\_type defined in PlayList(), as shown in drawing 33.

[0172]OUT\_time is 32 bit fields and stores the reproduction finish time of PlayItem. The semantics of OUT\_time changes with CPI\_type defined in PlayList(), as shown in drawing 34.

[0173]Connection\_Condition is the 2-bit field which shows the connected state between PlayItem to precede as shown in drawing 35, and the present PlayItem.

Drawing 36 is a figure explaining each state of Connection\_Condition shown in drawing 35.

[0174]Next, BridgeSequenceInfo is explained with reference to drawing 37.

BridgeSequenceInfo() is the attached information of the present PlayItem and has the information shown below. Bridge\_Clip\_Information\_file\_name which specifies a Bridge-Clip AV stream file and Clip Information file (drawing 45) corresponding to it is included.

[0175]It is an address of the source packet on Clip AV stream which PlayItem to precede refers to, and the source packet of the beginning of a Bridge-Clip AV stream file is connected following this source packet. This address is called

RSPN\_exit\_from\_previous\_Clip. It is an address of the source packet on Clip AV stream which the further present PlayItem refers to, and the source packet of the last of a Bridge-Clip AV stream file is connected before this source packet. This address is called RSPN\_enter\_to\_current\_Clip.

[0176]In drawing 37, RSPN\_arrival\_time\_discontinuity shows the address of the source packet which has a break point of arrival time base in a the Bridge-Clip AVstream file. This address is defined in ClipInfo() (drawing 46).

[0177]Drawing 38 is a figure showing the syntax of BridgeSequenceinfo. The syntax of BridgeSequenceinfo shown in drawing 38 to explain the field of Bridge\_Clip\_Information\_file\_name, The file name of Clip Information file corresponding to a Bridge-Clip AV stream file is shown. Clip\_stream\_type defined in ClipInfo() of this Clip Information file must show 'Bridge-Clip AV stream'.

[0178]32 bit fields of RSPN\_exit\_from\_previous\_Clip, It is a relative address of the source packet on Clip AV stream which PlayItem to precede refers to, and the source packet of the beginning of a Bridge-Clip AV stream file is connected following this source packet. RSPN\_exit\_from\_previous\_Clip, It is a size which makes a source packet number a unit, and the value of offset\_SPN defined in ClipInfo() from the source packet of the beginning of the Clip AV stream file which PlayItem to precede refers to is counted as an initial value.

[0179]32 bit fields of RSPN\_enter\_to\_current\_Clip, It is a relative address of the source packet on Clip AV stream which the present PlayItem refers to, and the source packet of the last of a Bridge-Clip AV stream file is connected before this source packet. RSPN\_exit\_from\_previous\_Clip, It is a size which makes a source packet number a unit, and the value of offset\_SPN defined in ClipInfo() from the source packet of the beginning of the Clip AV stream file which the present PlayItem refers to is counted as an initial value.

[0180]Next, SubPlayItem is explained with reference to drawing 39. Use of SubPlayItem() is allowed only when CPI\_type of PlayList() is EP\_map type. In this embodiment, SubPlayItem presupposes that it is used only for the purpose of postrecording of an audio. SubPlayItem() contains the data shown below. First, Clip\_information\_file\_name for specifying Clip which sub path in PlayList refers to is included.

[0181]SubPath\_IN\_time and SubPath\_OUT\_time for specifying the reproducing section of sub path in Clip are included. sync\_PlayItem\_id and sync\_start\_PTS\_of\_PlayItem for specifying the time in which sub path carries out a reproduction start on the time-axis of main path are included. Clip AV stream of the audio referred to at sub path must

not contain an STC break point (break point of system time base). The clock of the audio sample of Clip used for sub path is locked by the clock of the audio sample of main path.

[0182]Drawing 40 is a figure showing the syntax of SubPlayItem. The syntax of SubPlayItem shown in drawing 40 to explain the field of Clip\_Information\_file\_name, The file name of Clip Information file is shown and it is used by sub path in PlayList. Clip\_stream\_type defined in ClipInfo() of this Clip Information file must show Clip AV stream.

[0183]The 8-bit field of SubPath\_type shows the type of sub path. Here, as shown in drawing 41, only '0x00' is set up but other values are secured for the future.

[0184]The 8-bit field of sync\_PlayItem\_id shows PlayItem\_id of PlayItem in which the time in which sub path carries out a reproduction start on the time-axis of main path is contained. The value of PlayItem\_id corresponding to predetermined PlayItem is defined in PlayList() (refer to drawing 25).

[0185]The 32-bit field of sync\_start\_PTS\_of\_PlayItem, The time in which sub path carries out a reproduction start on the time-axis of main path is shown, and top 32 bits of PTS (Presentation Time Stamp) on PlayItem referred to by sync\_PlayItem\_id are shown. 32 bit fields of SubPath\_IN\_time store the reproduction start time of Sub path. SubPath\_IN\_time shows top 32 bits of PTS of 33 bit length corresponding to the first presentation unit in Sub Path.

[0186]32 bit fields of SubPath\_OUT\_time store the reproduction finish time of Sub path. SubPath\_OUT\_time shows top 32 bits of the value of Presentation\_end\_TS computed by a following formula.

$$\text{Presentation\_end\_TS} = \text{PTS\_out} + \text{AU\_duration}$$
 -- here, PTS\_out is PTS of 33 bit length corresponding to the presentation unit of the last of SubPath. AU\_duration is a display period of the 90-kHz unit of the presentation unit of the last of SubPath.

[0187]Next, PlayListMark() in the syntax of xxxxx.rpls shown in drawing 23 and yyyyy.vpls is explained. The mark information about PlayList is stored in this PlayListMark. Drawing 42 is a figure showing the syntax of PlayListMark. They are four character characters in which version\_number shows the version number of this PlayListMark() for explaining the syntax of PlayListMark shown in drawing 42. version\_number must be coded with "0045" according to ISO 646.

[0188]length is a 32-bit unsigned integer which shows the number of bytes of PlayListMark() from immediately after this length field to the last of PlayListMark(). number\_of\_PlayList\_marks is a 16-bit unsigned integer which shows the number of the mark currently stored in PlayListMark. number\_of\_PlayList\_marks may be 0. mark\_type

is the 8-bit field which shows the type of a mark, and is coded according to the table shown in [drawing 43](#).

[0189]32 bit fields of mark\_time\_stamp store the time stamp in which the point as which the mark was specified is shown. The semantics of mark\_time\_stamp changes with CPI\_type defined in PlayList(), as shown in [drawing 44](#). PlayItem\_id is the 8-bit field which specifies PlayItem on which the mark is put. The value of PlayItem\_id corresponding to predetermined PlayItem is defined in PlayList() (refer to [drawing 25](#)).

[0190]The 8-bit field of character\_set shows the encoding method of the character character coded by the mark\_name field. The encoding method corresponds to the value shown in [drawing 19](#). Eight bit fields of name\_length show the byte length of the mark name shown in the Mark\_name field. The field of mark\_name shows the name of a mark. The number of bytes of the left in this field to a name\_length number is an effective character character, and it shows the name of a mark. As for the value after these effective character character, what kind of value may be set up in the Mark\_name field.

[0191]The field of ref\_thumbnail\_index shows the information on the thumbnail image added to a mark. In the case of the value whose ref\_thumbnail\_index field is not 0xFFFF, the thumbnail image is added to the mark and the thumbnail image is stored in a mark.thmb file. The picture is referred to using the value of ref\_thumbnail\_index in a mark.thmb file (after-mentioned). When the ref\_thumbnail\_index field is 0xFFFF, it is shown that the thumbnail image is not added to the mark.

[0192]Next, Clip information file is explained. zzzzz.clpi (Clip information file file) comprises six objects, as shown in [drawing 45](#). They are ClipInfo(), STC\_Info(), ProgramInfo(), CPI(), ClipMark(), and MakersPrivateData(). "zzzzz" of the digit string with same AV stream (Clip AV stream or Bridge-Clip AV stream) and Clip Information file corresponding to it is used.

[0193]to explain the syntax of zzzzz.clpi (Clip information file file) shown in [drawing 45](#), ClipInfo\_Start\_address shows the start address of ClipInfo() by making the relative number of bytes from the byte of the head of a zzzzz.clpi file into a unit. A relative number of bytes is counted from zero.

[0194]STC\_Info\_Start\_address shows the start address of STC\_Info() by making the relative number of bytes from the byte of the head of a zzzzz.clpi file into a unit. A relative number of bytes is counted from zero. ProgramInfo\_Start\_address shows the start address of ProgramInfo() by making the relative number of bytes from the byte of the head of a zzzzz.clpi file into a unit. A relative number of bytes is counted from zero. CPI\_Start\_address shows the start address of CPI() by making the relative



number of bytes from the byte of the head of a zzzzz.clpi file into a unit. A relative number of bytes is counted from zero.

[0195]ClipMark\_Start\_address shows the start address of ClipMark() by making the relative number of bytes from the byte of the head of a zzzzz.clpi file into a unit. A relative number of bytes is counted from zero. MakersPrivateData\_Start\_address shows the start address of MakersPrivateData () by making the relative number of bytes from the byte of the head of a zzzzz.clpi file into a unit. A relative number of bytes is counted from zero. padding\_word (padding word) is inserted according to the syntax of a zzzzz.clpi file. N1, N2, N3, N4, and N5 must be zero or arbitrary positive integers. Any value may be made to be taken each padding word.

[0196]Next, ClipInfo is explained. Drawing 46 is a figure showing the syntax of ClipInfo. ClipInfo() stores the attribution information of the AV stream file (a Clip AV stream or a Bridge-Clip AV stream file) corresponding to it.

[0197]They are four character characters in which version\_number shows the version number of this ClipInfo() for explaining the syntax of ClipInfo shown in drawing 46. version\_number must be coded with "0045" according to ISO 646. length is a 32-bit unsigned integer which shows the number of bytes of ClipInfo() from immediately after this length field to the last of ClipInfo(). The 8-bit field of Clip\_stream\_type shows the type of the AV stream corresponding to a Clip Information file, as shown in drawing 47. The stream type of each type of AV stream is mentioned later.

[0198]The 32-bit field of offset\_SPN gives the offset value of the source packet number about the source packet of the beginning of an AV stream (Clip AV stream or Bridge-Clip AV stream) file. This offset\_SPN must be 0 when an AV stream file is first recorded on a disk.

[0199]As shown in drawing 48, when the first portion of an AV stream file is eliminated by edit, offset\_SPN is very good in values other than zero. The relative source packet number (relative address) which refers to offset\_SPN in this embodiment is often RSPN\_xxx (xxx changes.). It is described by in the form of example .RSPN\_EP\_start in syntax. A relative source packet number is a size which makes a source packet number a unit, and counts the value of offset\_SPN as an initial value from the source packet of the beginning of an AV stream file.

[0200]The number (SPN\_xxx) of the source packets to the source packet referred to by a relative source packet number from the source packet of the beginning of an AV stream file is computed with a following formula.

offset\_SPN shows  $SPN_{xxx} = RSPN_{xxx} - offset\_SPN$  drawing 48 the example in the case of being 4.

[0201]TS\_recording\_rate is a 24-bit unsigned integer -- this value -- a DVR drive (writing part 22) -- or the bit rate of required input and output of the AV stream from a DVR drive (read section 28) is given. record\_time\_and\_date, It is the 56-bit field in which time when the AV stream corresponding to Clip is recorded is stored, and 14 numbers are coded by 4-bit Binary Coded Decimal (BCD) about a /part / second at the time of year / moon / day/. For example, 2001/12/23:01:02:03 are coded with "0x20011223010203."

[0202]duration is the 24-bit field which showed the total reproduction time of Clip in the unit of time / part / second based on an arrival time clock. This field codes six numbers by 4-bit Binary Coded Decimal (BCD). For example, 01:45:30 is coded with "0x014530."

[0203]The flag of time\_controlled\_flag shows the recording mode of an AV stream file. When this time\_controlled\_flag is 1, the recording mode must fulfill the conditions which show that it is the mode in which it is recorded to the time progress after recording as a file size is proportional, and are shown in a following formula.

$$TS\_average\_rate * 192 / 188 * (t - start\_time) - \alpha \leq size\_clip \text{ -- } (t) \leq$$

$$TS\_average\_rate * 192 / 188 * (t - start\_time) + \alpha \text{ -- here, } TS\_average\_rate \text{ expresses the average bit rate of the transport stream of an AV stream file with a bytes/second unit.}$$

[0204]In an upper type, t shows the time expressed with a second bit, and start\_time is time when the source packet of the beginning of an AV stream file is recorded, and is expressed with a second bit. When the size of size\_clip (t) and the AV stream file in the time t is expressed per byte, for example, ten source packets are recorded by the time t from start\_time, size\_clip (t) is 10\*192 byte. alpha is a constant depending on TS\_average\_rate.

[0205]When time\_controlled\_flag is set to zero, not controlling the recording mode so that the file size of an AV stream is proportional to time progress of record is shown. For example, this is a case where transparent record of the input transport stream is carried out.

[0206]When, as for TS\_average\_rate, time\_controlled\_flag is set to one, this 24-bit field shows the value of TS\_average\_rate used by the upper formula. When time\_controlled\_flag is set to zero, this field has no meanings but must be set to 0. For example, the transport stream of a Variable Bit Rate is coded by the procedure shown below. A transformer portrait is first set to the value of TS\_recording\_rate. Next, a video stream is coded with a Variable Bit Rate. And a transport packet is intermittently coded by not using null packets.

[0207]32 bit fields of RSPN\_arrival\_time\_discontinuity are the relative addresses of the place which the discontinuity of arrival time base generates on a Bridge-ClipAV stream file. RSPN\_arrival\_time\_discontinuity, It is a size which makes a source packet number a unit, and the value of offset\_SPN defined in ClipInfo() from the source packet of the beginning of a Bridge-Clip AV stream file is counted as an initial value. The absolute address in the inside of the Bridge-Clip AV stream file is computed based on  $SPN_{xxx} = RSPN_{xxx} - \text{offset\_SPN}$  mentioned above.

[0208]The 144-bit field of reserved\_for\_system\_use is reserved for systems. When the flag of is\_format\_identifier\_valid is 1, it is shown that the field of format\_identifier is effective. When the flag of is\_original\_network\_ID\_valid is 1, it is shown that the field of original\_network\_ID is effective. When the flag of is\_transport\_stream\_ID\_valid is 1, it is shown that the field of transport\_stream\_ID is effective. When the flag of is\_service\_ID\_valid is 1, it is shown that the field of service\_ID is effective.

[0209]When the flag of is\_country\_code\_valid is 1, it is shown that the field of country\_code is effective. 32 bit fields of format\_identifier show the value of format\_identifier which registration descriptor (it defines as ISO/IEC 13818-1) has in a transport stream. 16 bit fields of original\_network\_ID show the value of original\_network\_ID defined in the transport stream. 16 bit fields of transport\_stream\_ID show the value of transport\_stream\_ID defined in the transport stream.

[0210]16 bit fields of service\_ID show the value of service\_ID defined in the transport stream. The 24-bit field of country\_code shows the country code defined by ISO3166. Each character character is coded by ISO8859-1. For example, Japan is expressed as "JPN" and coded with "0x4A 0x50 0x4E." stream\_format\_name is 16 character codes of ISO-646 which shows the name of the format organization which is carrying out the stream definition of the transport stream. The invalid byte in this field, value '0xFF' is set.

[0211]format\_identifier, original\_network\_ID, transport\_stream\_ID, service\_ID, country\_code, and stream\_format\_name, The service provider of a transport stream is shown and by this, Coding restrictions of an audio or a video stream and the stream definition of the standard of SI (service information) or private data streams other than an audio video stream can be recognized. These information can be used, in order that a decoder may perform initial setting of a decoder system before a decoding start whether the stream can be decoded and when it can decode and.

[0212]Next, STC\_Info is explained. In MPEG-2 transport stream, here call STC\_sequence the time intervals which do not contain the break point (break point of

system time base) of STC, and in Clip, STC\_sequence is specified with the value of STC\_sequence\_id. Drawing 50 is a figure explaining the STC section [ \*\*\*\* ]. The value of the STC same in the same STC\_sequence never appears (however, the maximum time length of Clip is restricted so that it may mention later). Therefore, the value of the same PTS also never appears in the same STC\_sequence. When an AV stream contains the STC break point of N ( $N > 0$ ) individual, the system time base of Clip is divided into STC\_sequence of an individual (N+1).

[0213]STC\_Info stores the address of the place which the discontinuity (discontinuity of system time base) of STC generates. So that it may explain with reference to drawing 51 RSPN\_STC\_start, The address is shown and k-th STC\_sequence ( $k \geq 0$ ) except the last STC\_sequence, It begins from the time when the source packet referred to by k-th RSPN\_STC\_start arrived, and finishes with the time when the source packet referred to by RSPN\_STC\_start of eye watch (k+1) arrived. The last STC\_sequence begins from the time when the source packet referred to by the last RSPN\_STC\_start arrived, and is ended at the time when the last source packet arrived.

[0214]Drawing 52 is a figure showing the syntax of STC\_Info. They are four character characters in which version\_number shows the version number of this STC\_Info() for explaining the syntax of STC\_Info shown in drawing 52. version\_number must be coded with "0045" according to ISO 646.

[0215]length is a 32-bit unsigned integer which shows the number of bytes of STC\_Info() from immediately after this length field to the last of STC\_Info(). When CPI\_type of CPI() shows TU\_map type, this length field may set zero. When CPI\_type of CPI() shows EP\_map type, num\_of\_STC\_sequences must be one or more values.

[0216]The 8-bit unsigned integer of num\_of\_STC\_sequences shows the number of STC\_sequence in the inside of Clip. This value shows the loop count of for-loop following this field. STC\_sequence\_id corresponding to predetermined STC\_sequence is defined in for-loop containing RSPN\_STC\_start by the turn that RSPN\_STC\_start corresponding to the STC\_sequence appears. STC\_sequence\_id is started from 0.

[0217]32 bit fields of RSPN\_STC\_start show the address which STC\_sequence starts on an AV stream file. RSPN\_STC\_start shows the address which the break point of system time base generates in an AV stream file. RSPN\_STC\_start is good also as a relative address of a source packet which has PCR of the beginning of new system time base in an AV stream. RSPN\_STC\_start is a size which makes a source packet number a unit, and counts the value of offset\_SPN defined in ClipInfo() from the source packet of the beginning of an AV stream file as an initial value. The absolute address in the inside of the AV stream file is computed by  $SPN_{xxx} = RSPN_{xxx} -$

offset\_SPN already mentioned above.

[0218]Next, ProgramInfo in the syntax of zzzzz.clip shown in drawing 45 is explained.

The time intervals which have the following feature in Clip are called program\_sequence for explaining here, referring to drawing 53. First, the value of PCR\_PID does not change. Next, the number of video elementary streams does not change. The encoded information defined by the value and VideoCodingInfo of PID about each video stream does not change. The number of audio elementary streams does not change. The encoded information defined by the value and AudioCodingInfo of PID about each audio stream does not change.

[0219]In the same time, program\_sequence has only one system time base. In the same time, program\_sequence has only one PMT. ProgramInfo() stores the address of the place which program\_sequence starts. RSPN\_program\_sequence\_start shows the address.

[0220]Drawing 54 is a figure showing the syntax of ProgramInfo. They are four character characters in which version\_number shows the version number of this ProgramInfo() for explaining SHINTAKU of ProgramInfo shown in drawing 54. version\_number must be coded with "0045" according to ISO 646.

[0221]length is a 32-bit unsigned integer which shows the number of bytes of ProgramInfo() from immediately after this length field to the last of ProgramInfo(). When CPI\_type of CPI() shows TU\_map type, this length field may be set to zero. When CPI\_type of CPI() shows EP\_map type, number\_of\_programs must be one or more values.

[0222]The 8-bit unsigned integer of number\_of\_program\_sequences shows the number of program\_sequence in the inside of Clip. This value shows the loop count of for-loop following this field. When program\_sequence does not change in Clip, number\_of\_program\_sequences must have one set. 32 bit fields of RSPN\_program\_sequence\_start are the relative addresses of the place which a program sequence starts on an AV stream file.

[0223]RSPN\_program\_sequence\_start is a size which makes a source packet number a unit, and counts the value of offset\_SPN defined in ClipInfo() from the source packet of the beginning of an AV stream file as an initial value. The absolute address in the inside of the AV stream file is computed by  $SPN_{xxx} = RSPN_{xxx} - offset\_SPN$ . The RSPN\_program\_sequence\_start value must appear in an ascending order in for-loop of syntax.

[0224]16 bit fields of PCR\_PID show PID of a transport packet including the PCR field effective in the program\_sequence. Eight bit fields of number\_of\_videos show the loop

count of video\_stream\_PID and for-loop containing VideoCodingInfo(). Eight bit fields of number\_of\_audios show the loop count of audio\_stream\_PID and for-loop containing AudioCodingInfo(). 16 bit fields of video\_stream\_PID show PID of the transport packet containing a video stream effective in the program\_sequence. VideoCodingInfo() following this field must explain the contents of the video stream referred to by that video\_stream\_PID.

[0225]16 bit fields of audio\_stream\_PID show PID of the transport packet containing an audio stream effective in the program\_sequence. AudioCodingInfo() following this field must explain the contents of the video stream referred to by that audio\_stream\_PID.

[0226]The turn that the value of video\_stream\_PID appears in for-loop of syntax must be equal to the turn that PID of the video stream is coded in PMT effective in the program\_sequence. The turn that the value of audio\_stream\_PID appears in for-loop of syntax must be equal to the turn that PID of the audio stream is coded in PMT effective in the program\_sequence.

[0227]Drawing 55 is a figure showing the syntax of VideoCodingInfo in the syntax of ProgramInfo shown in drawing 54. For explaining the syntax of VideoCodingInfo shown in drawing 55, eight bit fields of video\_format show the format video corresponding to video\_stream\_PID in ProgramInfo(), as shown in drawing 56.

[0228]Eight bit fields of frame\_rate show the frame rate of the video corresponding to video\_stream\_PID in ProgramInfo(), as shown in drawing 57. Eight bit fields of display\_aspect\_ratio show the display aspect ratio of the video corresponding to video\_stream\_PID in ProgramInfo(), as shown in drawing 58.

[0229]Drawing 59 is a figure showing the syntax of AudioCodingInfo in the syntax of ProgramInfo shown in drawing 54. For explaining the syntax of AudioCodingInfo shown in drawing 59, eight bit fields of audio\_coding show the encoding method of the audio corresponding to audio\_stream\_PID in ProgramInfo(), as shown in drawing 60.

[0230]Eight bit fields of audio\_component\_type show the component type of the audio corresponding to audio\_stream\_PID in ProgramInfo(), as shown in drawing 61. Eight bit fields of sampling\_frequency show the sampling frequency of the audio corresponding to audio\_stream\_PID in ProgramInfo(), as shown in drawing 62.

[0231]Next, CPI (Characteristic Point Information) in the syntax of zzzzz.clip shown in drawing 45 is explained. Since the hour entry in an AV stream and the address in the file are associated, there is CPI. There are two types of CPI(s) and they are EP\_map and TU\_map. As shown in drawing 63, when CPI\_type in CPI() is EP\_map type, the CPI() contains EP\_map. As shown in drawing 64, when CPI\_type in CPI() is TU\_map

type, the CPI() contains TU\_map. One AV stream has one EP\_map or one TU\_map. When an AV stream is a SESF transport stream, Clip corresponding to it must have EP\_map.

[0232]Drawing 65 is a figure showing the syntax of CPI. They are four character characters in which version\_number shows the version number of this CPI() for explaining the syntax of CPI shown in drawing 65. version\_number must be coded with "0045" according to ISO 646. length is a 32-bit unsigned integer which shows the number of bytes of CPI() from immediately after this length field to the last of CPI(). As shown in drawing 66, CPI\_type is a 1-bit flag and expresses the type of CPI of Clip. [0233]Next, EP\_map in the syntax of CPI shown in drawing 65 is explained. There are two types of EP\_map and it is EP\_map for video streams, and EP\_map for audio streams. EP\_map\_type in EP\_map distinguishes the type of EP\_map. When Clip contains one or more video streams, EP\_map for video streams must be used. When Clip does not contain a video stream but contains one or more audio streams, EP\_map for audio streams must be used.

[0234]EP\_map for video streams is explained with reference to drawing 67. EP\_map for video streams has data called stream\_PID, PTS\_EP\_start, and RSPN\_EP\_start. stream\_PID shows PID of the transport packet which transmits a video stream. PTS\_EP\_start shows PTS of the access unit which begins from the sequence header of a video stream. RSPN\_EP\_start shows the address of the source packet containing the 1st byte of the access unit referred to by PTS\_EP\_start in an AV stream.

[0235]The sub table called EP\_map\_for\_one\_stream\_PID() is made for every video stream transmitted by the transport packet with the same PID. When two or more video streams exist in Clip, EP\_map may also contain two or more EP\_map\_for\_one\_stream\_PID().

[0236]EP\_map for audio streams has data called stream\_PID, PTS\_EP\_start, and RSPN\_EP\_start. stream\_PID shows PID of the transport packet which transmits an audio stream. PTS\_EP\_start shows PTS of the access unit of an audio stream. RSPN\_EP\_start shows the address of the source packet containing the 1st byte of the access unit referred to by PTS\_EP\_start in an AV stream.

[0237]The sub table called EP\_map\_for\_one\_stream\_PID() is made for every audio stream transmitted by the transport packet with the same PID. When two or more audio streams exist in Clip, EP\_map may also contain two or more EP\_map\_for\_one\_stream\_PID().

[0238]One EP\_map\_for\_one\_stream\_PID() is made by explaining the relation between EP\_map and STC\_Info on one table regardless of the break point of STC. By comparing

the value of RSPN\_STC\_start defined in the value of RSPN\_EP\_start and STC\_Info() shows the boundary of the data of EP\_map belonging to each STC\_sequence (see drawing 68). – EP\_map must have one EP\_map\_for\_one\_stream\_PID to the range of the continuous stream transmitted by the same PID. When shown in drawing 69, although program#1 and program#3 have the same video PID, since the data range is not continuing, they must have EP\_map\_for\_one\_stream\_PID for every program.

[0239]Drawing 70 is a figure showing the syntax of EP\_map. For explaining the syntax of EP\_map shown in drawing 70, EP\_type is the 4-bit field, and as shown in drawing 71, it shows the entry point type of EP\_map. EP\_type shows the semantics of the data field following this field. EP\_type must be set to zero ('video') when Clip contains one or more video streams. Or EP\_type must be set to one ('audio'), when Clip does not contain a video stream but contains one or more audio streams.

[0240]The 16-bit field of number\_of\_stream\_PIDs shows the loop count of for-loop which has number\_of\_stream\_PIDs in EP\_map() in a variable. The 16-bit field of stream\_PID (k), PID of the transport packet which transmits the k-th elementary stream (video or audio stream) referred to by EP\_map\_for\_one\_stream\_PID (num\_EP\_entries (k)) is shown. case EP\_type is equal to zero ('video') -- the elementarist ream -- a video stream -- kicking does not become impossible. When EP\_type is equal to one ('audio'), the elementarist ream must be an audio stream.

[0241]The 16-bit field of num\_EP\_entries (k) shows num\_EP\_entries (k) referred to by EP\_map\_for\_one\_stream\_PID (num\_EP\_entries (k)).

EP\_map\_for\_one\_stream\_PID\_Start\_address (k): This 32-bit field, The relative byte position from which EP\_map\_for\_one\_stream\_PID (num\_EP\_entries (k)) begins in EP\_map() is shown. This value is shown by the size from the 1st byte of EP\_map().

[0242]padding\_word must be inserted according to the syntax of EP\_map(). X and Y must be zero or arbitrary positive integers. Each padding word may take any value.

[0243]Drawing 72 is a figure showing the syntax of EP\_map\_for\_one\_stream\_PID. The semantics of the 32-bit field of PTS\_EP\_start changes with EP\_type defined in EP\_map() to explain the syntax of EP\_map\_for\_one\_stream\_PID shown in drawing 72. When EP\_type is equal to zero ('video'), this field has top 32 bits of PTS of the 33-bit accuracy of the access unit which starts with the sequence header of a video stream. When EP\_type is equal to one ('audio'), this field has top 32 bits of PTS of the 33-bit accuracy of the access unit of an audio stream.

[0244]The semantics of the 32-bit field of RSPN\_EP\_start changes with EP\_type defined in EP\_map(). When EP\_type is equal to zero ('video'), this field shows the relative address of the source packet containing the 1st byte of the sequence header



of the access unit referred to by PTS\_EP\_start in an AV stream. Or when EP\_type is equal to one ('audio'), this field shows the relative address of the source packet containing the first byte of the audio frame of the access unit referred to by PTS\_EP\_start in an AV stream.

[0245]RSPN\_EP\_start is a size which makes a source packet number a unit, and counts the value of offset\_SPN defined in ClipInfo() from the source packet of the beginning of an AV stream file as an initial value. The absolute address in the inside of the AV stream file is computed by  $SPN_{xxx} = RSPN_{xxx} - offset\_SPN$ . The value of RSPN\_EP\_start must appear in an ascending order in for-loop of syntax.

[0246]Next, TU\_map is explained with reference to drawing 73. TU\_map makes one time-axis based on the arrival time clock (clock of an arrival time base) of a source packet. The time-axis is called TU\_map\_time\_axis. The starting point of TU\_map\_time\_axis is shown by offset\_time in TU\_map(). TU\_map\_time\_axis is divided into a fixed unit from offset\_time. The unit is called time\_unit.

[0247]In each time\_unit in an AV stream, the address on the AV stream file of the source packet of the first perfect form is stored in TU\_map. These addresses are called RSPN\_time\_unit\_start. In a TU\_map\_time\_axis top, it is k. The time when time\_unit of eye watch ( $k \geq 0$ ) begins is called TU\_start\_time (k). This value is computed based on a following formula.

$TU\_start\_time(k) = offset\_time + k * time\_unit\_size$   
TU\_start\_time (k) has the accuracy of 45 kHz.

[0248]Drawing 74 is a figure showing the syntax of TU\_map. The field of the 32-bit length of offset\_time gives the offset time to TU\_map\_time\_axis for explaining the syntax of TU\_map shown in drawing 74. This value shows the offset time to time\_unit of the beginning in Clip. offset\_time is a size which makes a unit the 45-kHz clock drawn from the arrival time clock of 27-MHz accuracy. offset\_time must be set to zero when an AV stream is recorded as new Clip.

[0249]32 bit fields of time\_unit\_size give the size of time\_unit, and it is a size which makes a unit the 45-kHz clock drawn from the arrival time clock of 27-MHz accuracy. time\_unit\_size is good to use 1 or less ( $time\_unit\_size \leq 45000$ ) second. 32 bit fields of number\_of\_time\_unit\_entries show the number of entries of time\_unit currently stored in TU\_map().

[0250]32 bit fields of RSPN\_time\_unit\_start show the relative address of the place which each time\_unit starts in an AV stream. RSPN\_time\_unit\_start is a size which makes a source packet number a unit, and counts the value of offset\_SPN defined in ClipInfo() from the source packet of the beginning of an AV stream file as an initial

value. The absolute address in the inside of the AV stream file is computed by  $SPN\_xxx = RSPN\_xxx - offset\_SPN$ . The value of `RSPN_time_unit_start` must appear in an ascending order in for-loop of syntax. (k+1) When anything does not have a source packet in time\_unit of eye watch, `RSPN_time_unit_start` of eye watch (k+1) must be equal to k-th `RSPN_time_unit_start`.

[0251]ClipMark in the syntax of `zzzzz.clip` shown in [drawing 45](#) is explained. ClipMark is the mark information about a clip and is stored into ClipMark. This mark is set by a recorder (recording and reproducing device 1), and is not set by the user.

[0252][Drawing 75](#) is a figure showing the syntax of ClipMark. They are four character characters in which `version_number` shows the version number of this ClipMark() for explaining the syntax of ClipMark shown in [drawing 75](#). `version_number` must be coded with "0045" according to ISO 646.

[0253]`length` is a 32-bit unsigned integer which shows the number of bytes of ClipMark() from immediately after this `length` field to the last of ClipMark(). The 16-bit unsigned integer `number_of_Clip_marks` indicates the number of the mark currently stored in ClipMark to be. `number_of_Clip_marks` may be 0. `mark_type` is the 8-bit field which shows the type of a mark, and is coded according to the table shown in [drawing 76](#).

[0254]`mark_time_stamp` is 32 bit fields and stores the time stamp in which the point as which the mark was specified is shown. The semantics of `mark_time_stamp` changes with `CPI_type` in `PlayList()`, as shown in [drawing 77](#).

[0255]When, as for `STC_sequence_id`, `CPI_type` in `CPI()` shows EP\_map type, this 8-bit field shows `STC_sequence_id` of the STC continuation section on which `mark_time_stamp` is put. When `CPI_type` in `CPI()` shows TU\_map type, this 8-bit field has no meanings, but is set to zero. The 8-bit field of `character_set` shows the encoding method of the character character coded by the `mark_name` field. The encoding method corresponds to the value shown in [drawing 19](#).

[0256]Eight bit fields of `name_length` show the byte length of the mark name shown in the `Mark_name` field. The field of `mark_name` shows the name of a mark. The number of bytes of the left in this field to a `name_length` number is an effective character character, and it shows the name of a mark. In the `mark_name` field, what kind of value may be [ value after these effective character character ] contained.

[0257]The field of `ref_thumbnail_index` shows the information on the thumbnail image added to a mark. In the case of the value whose `ref_thumbnail_index` field is not 0xFFFF, the thumbnail image is added to the mark and the thumbnail image is stored in a `mark.thmb` file. The picture is referred to using the value of `ref_thumbnail_index` in

a mark.thmb file. When the ref\_thumbnail\_index field is 0xFFFF, the thumbnail image is not added to the mark.

[0258]Drawing 78 is a figure showing other syntax of ClipMark replaced with drawing 75, and drawing 79 shows the example of the table of mark\_type which can be set in that case and which is replaced with drawing 76. reserved\_for\_maker\_ID is the 16-bit field which shows maker ID of the maker which defines the mark\_type, when mark\_type shows the value of 0xC0 to 0xFF. A DVR format licenser specifies maker ID.

mark\_entry() is information which shows the point specified as the marking point, and the details of the syntax are mentioned later. representative\_picture\_entry() is information which shows the point of the picture representing the mark shown by mark\_entry(), and the details of the syntax are mentioned later.

[0259]When a user reproduces an AV stream, ClipMark is used in order to be able to search the contents visually. A DVR player uses GUI (graphical user interface), and shows a user the information on ClipMark. It is better to show the picture which representative\_picture\_entry() shows rather than the picture which mark\_entry() shows, in order to display the information on ClipMark visually.

[0260]The example of mark\_entry() and representative\_picture\_entry() is shown in drawing 80. For example, suppose that the program name (title) of the program is displayed after a certain program begins, and carrying out for a while (after several seconds). When making ClipMark, mark\_entry() is put on the start points of the program, and it may be made to put representative\_picture\_entry() on the point as which the program name (title) of the program is displayed.

[0261]If a DVR player displays the picture of representative\_picture\_entry on GUI and a user specifies the picture, a DVR player will start reproduction from the point on which mark\_entry was put.

[0262]The syntax of mark\_entry() and representative\_picture\_entry() is shown in drawing 81.

[0263]mark\_time\_stamp is 32 bit fields, In mark\_entry(), the time stamp in which the point as which the mark was specified is shown is stored, In representative\_picture\_entry(), the time stamp in which the point of the picture representing the mark shown by mark\_entry() is shown is stored.

[0264]Next, in order to specify ClipMark, the information on the time stamp base by PTS is not used, The example of the syntax of mark\_entry() in the case of using the information on an address base and representative\_picture\_entry() is shown in drawing 82.

[0265]In RSPN\_ref\_EP\_start and mark\_entry(), the relative address of the source

packet which shows the entry point of the stream for decoding the picture of a marking point in an AV stream is shown. In `representative_picture_entry()`, the relative address of the source packet which shows the entry point of the stream for decoding the picture representing the mark shown by `mark_entry()` is shown. The value of `RSPN_ref_EP_start`, It must be stored as `RSPN_EP_start` in `EP_map`, and the value of `PTS_EP_start` corresponding to the `RSPN_EP_start` must be the value nearest than `PTS` of the picture of a marking point among the past in `EP_map`.

[0266]`offset_num_pictures` is the 32-bit field and shows the number of pictures of offset to the picture shown with a marking point by a display order from the picture referred to by `RSPN_ref_EP_start`. This number is counted from zero. In the case of the example of [drawing 83](#), `offset_num_pictures` is set to six.

[0267]Next, in order to specify `ClipMark`, another example of the syntax of `mark_entry()` in the case of using the information on an address base and `representative_picture_entry()` is shown in [drawing 84](#).

[0268]In `mark_entry()`, `RSPN_mark_point` shows the relative address of the source packet which contains the 1st byte that is an access unit which the mark refers to in an AV stream. In `representative_picture_entry()`, the relative address of the source packet containing the 1st byte of the coding picture representing the mark shown by `mark_entry()` is shown.

[0269]`RSPN_mark_point` is a size which makes a source packet number a unit, and counts the value of `offset_SPN` defined in `Clip Information` file from the source packet of the beginning of an AV stream file as an initial value.

[0270]The relation between `ClipMark` and `EP_map` is explained using [drawing 85](#). In the case of this example, `EP_map` specifies `I0`, `I1`, and `In` as an address of an entry point, and assumes that `I` picture which follows a sequence header from these addresses has begun. What is necessary is to read data from `I1` which is the nearest entry point before the address of `M1`, and just to start, in order to be able to decode the picture started from the source packet, when `ClipMark` specifies `M1` as an address of a certain mark.

[0271]Since `MakersPrivateData` was already explained with reference to [drawing 22](#), the explanation is omitted.

[0272]Next, the thumbnail information (Thumbnail Information) is explained. A thumbnail image is stored in a `menu.thmb` file or a `mark.thmb` file. These files are the same syntax structures and have only one `Thumbnail()`. A `menu.thmb` file stores a menu thumbnail image, i.e., the picture representing Volume, and the picture representing each `PlayList`. All the menu thumbnails are stored only in one `menu.thmb`

file.

[0273]A mark.thmb file stores the picture showing a mark thumbnail image, i.e., a marking point. All the mark thumbnails to all the Playlist(s) and Clip(s) are stored only in one mark.thmb file. Since a thumbnail is added and deleted frequently, add operation and operation of partial deletion must be able to be performed at high speed easily. Thumbnail() has a block structure for this reason. The data of a picture is divided into some portions and each portion is stored in one tn\_block. One image data is stored in tn\_block which \*\*\*\*\*ed). tn\_block which is not used may exist in the sequence of tn\_block. The byte length of one thumbnail image is variable.

[0274]Drawing 86 is a figure showing the syntax of menu.thmb and mark.thmb, and drawing 87 is a figure showing the syntax of Thumbnail in the syntax of menu.thmb shown in drawing 86, and mark.thmb. They are four character characters in which version\_number shows the version number of this Thumbnail() for explaining the syntax of Thumbnail shown in drawing 87. version\_number must be coded with "0045" according to ISO 646.

[0275]length is a 32-bit unsigned integer which shows the number of bytes of MakersPrivateData() from immediately after this length field to the last of Thumbnail(). tn\_blocks\_start\_address is a 32-bit unsigned integer which shows the head byte address of the first tn\_block by making the relative number of bytes from the byte of the head of Thumbnail() into a unit. A relative number of bytes is counted from zero. number\_of\_thumbnails is a 16-bit unsigned integer which gives the number of entries of the thumbnail image contained in Thumbnail().

[0276]tn\_block\_size is a 16-bit unsigned integer which gives the size of one tn\_block by making 1024 bytes into a unit. For example, if it becomes tn\_block\_size=1, it shows that the size of one tn\_block is 1024 bytes. number\_of\_tn\_blocks is a 116-bit unsigned integer showing the number of entries of tn\_block in this Thumbnail(). thumbnail\_index is a 16-bit unsigned integer showing the index number of a thumbnail image expressed with the thumbnail information on the "for" loop batch which begins from this thumbnail\_index field. Don't use a value called 0xFFFF as thumbnail\_index. Refer to thumbnail\_index for ref\_thumbnail\_index in UIAppInfoVolume(), UIAppInfoPlaylist(), PlaylistMark(), and ClipMark().

[0277]thumbnail\_picture\_format is an 8-bit unsigned integer showing the picture format of a thumbnail image, and takes a value as shown in drawing 88. DCF and PNG in front are allowed only within "menu.thmb." The mark thumbnail must take value "0x00" (MPEG-2 Video I-picture).

[0278]picture\_data\_size is a 32-bit unsigned integer which shows the byte length of a

thumbnail image per byte. start\_tn\_block\_number is a 16-bit unsigned integer showing the tn\_block number of tn\_block from which the data of a thumbnail image begins. The head of thumbnail image data must be in agreement with the head of tb\_block. A tn\_block number begins from 0 and is related to the value of the variable k in the for-loop of tn\_block.

[0279]x\_picture\_length is a 16-bit unsigned integer showing the horizontal number of pixels of the frame picture frame of a thumbnail image. y\_picture\_length is a 16-bit unsigned integer showing the number of pixels of the perpendicular direction of the frame picture frame of a thumbnail image. tn\_block is a field in which a thumbnail image is stored. All the tn\_block in Thumbnail() is the same sizes (fixed length), and the size is defined by tn\_block\_size.

[0280]Drawing 89 is the figure which meant typically how thumbnail image data would be stored in tn\_block. Like drawing 89, each thumbnail image data begins from the head of tn\_block, and, in the case of the size exceeding 1 tn\_block, it is stored using continuous following tn\_block. By doing in this way, the picture data which is variable length becomes possible [ managing as fixed-length data ], and can respond now by simple processing to edit called deletion.

[0281]Next, an AV stream file is explained. An AV stream file is stored in an "M2TS" directory (drawing 14). There are two types of AV stream files, and they are a Clip AV stream and a Bridge-Clip AV stream file. It must be the structure of a DVR MPEG-2 transport stream file where both AV streams are defined henceforth [ this ].

[0282]First, DVR MPEG-2 A transport stream is explained. The structure of DVR MPEG-2 transport stream is shown in drawing 90. An AV stream file has the structure of a DVR MPEG2 transport stream. A DVR MPEG2 transport stream comprises Aligned unit of integer pieces. The size of Alignedunit is 6144. Byte (2048\*3 byte) It is. Aligned unit begins from the 1st byte of a source packet. A source packet is 192-byte length. One source packet comprises TP\_extra\_header and a transport packet. TP\_extra\_header is 4-byte length and a transport packet is 188-byte length.

[0283]One Aligned unit comprises 32 source packets. Aligned unit of the last in a DVR MPEG2 transport stream also comprises 32 source packets. Therefore, the termination of the DVR MPEG2 transport stream is carried out on the boundary of Aligned unit. When the number of the transport packets of the input transport stream recorded on a disk is not a multiple of 32, a source packet with null packets (transport packet of PID=0x1FFF) must be used for the last Aligned unit. The file system must not add excessive information to a DVR MPEG2 transport stream.

[0284]The recorder model of DVR MPEG-2 transport stream is shown in drawing 91.

The recorder shown in drawing 91 is a model on the concept for specifying a recording process. DVR MPEG-2 transport stream follows this model.

[0285]The input timing of MPEG-2 transport stream is explained. An input MPEG2 transport stream is a full transport stream or a partial transport stream. The MPEG2 transport stream inputted must follow ISO/IEC 13818-1 or ISO/IEC 13818-9. The  $i$ -th byte of an MPEG2 transport stream, It is simultaneously inputted into T-STD(it is prescribed by ISO/IEC 13818-1 Transport stream system target decoder) 51 and saw spa KETTAIZA (source packetizer)54 at time  $t(i)$ .  $R_{pk}$  is the instant maximum of the input rate of a transport packet.

[0286]27MHz PLL52 generates the frequency of 27 MHz clocks. The frequency of 27 MHz clocks is locked by the value of PCR (Program Clock Reference) of MPEG-2 transport stream. Arrival time clock counter (arrival time clock counter)53 is a binary counter which counts a pulse with a frequency of 27 MHz.  $Arrival\_time\_clock(i)$  is the counted value of arrival time clockcounter53 in time  $t(i)$ .

[0287]source packetizer54 adds TP\_extra\_header to all the transport packets, and makes a source packet.  $Arrival\_time\_stamp$  expresses the time when the 1st byte of a transport packet arrives to both T-STD51 and saw spa KETTAIZA 54.

$Arrival\_time\_stamp(k)$  is a sampled value of  $Arrival\_time\_clock(k)$ , as shown in a following formula, and  $k$  shows the 1st byte of a transport packet here.

$arrival\_time\_stamp(k) = arrival\_time\_clock(k) \% 2^{30}$ [0288]When the time interval of two transport packets inputted continuously becomes  $2^{30} /$  more than 27 million second (about 40 seconds), The difference of  $arrival\_time\_stamp$  of the two transport packets should be set as it has been  $2^{30} / 27$  million seconds. It prepares for the case where a recorder becomes such.

[0289]Smoothing buffer (smoothing buffer)55 carries out smoothing of the bit rate of an input transport stream. Don't overflow the smoothing buffer 55.  $R_{max}$  is the output bit rate of the source packet from the smoothing buffer 55 in case the smoothing buffer 55 is not empty. When the smoothing buffer 55 is empty, the output bit rate from the smoothing buffer 55 is zero.

[0290]Next, the parameter of the recorder model of DVR MPEG-2 transport stream is explained. A value called  $R_{max}$  is given by  $TS\_recording\_rate$  defined in  $ClipInfo()$  corresponding to an AV stream file. This value is computed by a following formula. The value of  $R_{max} = TS\_recording\_rate * 192 / 188$   $TS\_recording\_rate$  is a size which makes bytes/second a unit.

[0291]When an input transport stream is a SESF transport stream,  $R_{pk}$  must be equal to  $TS\_recording\_rate$  defined in  $ClipInfo()$  corresponding to an AV stream file. When an

input transport stream is not a SESEF transport stream, Refer to the value set and defined without a descriptor, for example, maximum\_bitrate\_descriptor, partial\_transport\_stream\_descriptor, etc. of MPEG-2 transport stream for this value.

[0292]When an input transport stream is a SESEF transport stream, the size (smoothing buffer size) of the smoothing buffer 55 is zero. When an input transport stream is not a SESEF transport stream, The size of the smoothing buffer 55 The descriptor of MPEG-2 transport stream, For example, the value defined in smoothing\_buffer\_descriptor, short\_smoothing\_buffer\_descriptor, partial\_transport\_stream\_descriptor, etc. may be referred to.

[0293]A record machine (recorder) and the reproduction machine (player) must prepare the buffer of sufficient size. Default buffer size is 1536 bytes.

[0294]Next, the player model of DVR MPEG-2 transport stream is explained. Drawing 92 is a figure showing the player model of DVR MPEG-2 transport stream. This is a model on the concept for specifying reconstructive processing. DVR MPEG-2 transport stream follows this model.

[0295]27 MHz X-tal(crystal oscillator) 61 generates the frequency of 27 MHz. The error span of 27-MHz frequency must be  $\pm 30$  ppm (27 million $\pm$  810 Hz). arrival time clock counter62 is a binary counter which counts a pulse with a frequency of 27 MHz. arrival\_time\_clock(i) is the counted value of arrival time clock counter62 in time t(i).

[0296]In smoothing buffer64, Rmax is the input bit rate of the source packet to the smoothing buffer 64 in case the smoothing buffer 64 is not full. When the smoothing buffer 64 is full, the input bit rate to the smoothing buffer 64 is zero.

[0297]to explain the output timing of MPEG-2 transport stream, When arrival\_time\_stamp of the present source packet is equal to the value which is 30 bits of LSB of arrival\_time\_clock(i), the transport packet of the source packet is drawn out from the smoothing buffer 64. Rpk is the instant maximum of a transport packet rate. Don't carry out underflow of the smoothing buffer 64.

[0298]About the parameter of the player model of DVR MPEG-2 transport stream, it is the same as that of the parameter of the recorder model of DVR MPEG-2 transport stream mentioned above.

[0299]Drawing 93 is a figure showing the syntax of Source packet. transport\_packet() is MPEG-2 transport packet specified by ISO/IEC 13818-1. The syntax of TP\_Extra\_header in the syntax of Source packet shown in drawing 93 is shown in drawing 94. It is an integer as which copy\_permission\_indicator expresses copy restrictions of the pay load of a transport packet for explaining the syntax of



TP\_Extra\_header shown in drawing 94. Copy restrictions can be set to copy free, no more copy, copy once, or copy prohibited. Drawing 95 shows the value of copy\_permission\_indicator, and the relation in the mode specified by them.

[0300]copy\_permission\_indicator is added to all the transport packets. When recording an input transport stream using an IEEE1394 digital interface, the value of copy\_permission\_indicator, It may relate with the value of EMI (Encryption Mode Indicator) in IEEE1394 isochronous packet header. The value of copy\_permission\_indicator may be related with the value of CCI embedded into the transport packet, when recording an input transport stream without using an IEEE1394 digital interface. The value of copy\_permission\_indicator may be related with the value of CGMS-A of an analog signal when carrying out self encoding of the analog signal input.

[0301]arrival\_time\_stamp is following formula  $\text{arrival\_time\_stamp}(k) = \text{arrival\_time\_clock}(k) \% 2^{30}$ , it is an integral value with the value specified by arrival\_time\_stamp.

[0302]The Clip AV stream must have [ defining a Clip AV stream and ] the structure of DVR MPEG-2 transport stream where a definition which was mentioned above is carried out. arrival\_time\_clock(i) must increase continuously in a Clip AV stream. Even if the break point of system time base (STC base) exists in a Clip AV stream, arrival\_time\_clock(i) of the Clip AV stream must increase continuously.

[0303]The maximum of the start in a Clip AV stream and the difference of arrival\_time\_clock(i) between ends must be 26 hours. This restriction guarantees that PTS (Presentation Time Stamp) of the same value never appears in a Clip AV stream, when the break point of system time base (STC base) does not exist in an MPEG2 transport stream. The MPEG 2 systems standard has specified the wrap around cycle of PTS as  $233 / 90000$  second (about 26.5 hours).

[0304]The Bridge-Clip AV stream must have [ defining a Bridge-Clip AV stream and ] the structure of DVR MPEG-2 transport stream where a definition which was mentioned above is carried out. The Bridge-Clip AV stream must contain the break point of one arrival time base. The transport stream before and behind the break point of arrival time base must follow DVR-STD which must follow restriction of the coding mentioned later and is mentioned later.

[0305]In this embodiment, the video between PlayItem(s) in edit and seamless connection of an audio are supported. Making between PlayItem seamless connection guarantees "the continuous supply of data", and "seamless decoding processing" to a player/recorder. "The continuous supply of data" is being able to guarantee a file

system supplying data by the required bit rate so that a decoder's may not be made to cause the underflow of a buffer. The real time nature of data is guaranteed, and data is stored by the block unit which sufficient size followed so that data can be read from a disk.

[0306]"Seamless decoding processing" is that a player can display the audio video data recorded on the disk, without making the reproducing output of a decoder start a pause and a gap.

[0307]The AV stream which PlayItem by which seamless connection is made refers to is explained. It can be judged whether connection of PlayItem to precede and the present PlayItem is guaranteed to indicate by seamless from the connection\_condition field defined in the present PlayItem. The seamless connection between PlayItem(s) has the method of using Bridge-Clip, and a method which is not used.

[0308]Drawing 96 shows the relation between PlayItem preceded in the case of using Bridge-Clip, and the present PlayItem. In drawing 96, the stream data which a player reads give a shadow and are shown. TS1 shown in drawing 96 comprises the stream data which were able to attach the shadow of Clip1 (Clip AV stream), and the stream data which were able to attach the shadow before RSPN\_arrival\_time\_discontinuity of Bridge-Clip.

[0309]The stream data which were able to attach the shadow of Clip1 of TS1, From the address of a stream required in order to decode the presentation unit corresponding to IN\_time (illustrated by IN\_time1 in drawing 96) of PlayItem to precede, They are the stream data to the source packet referred to by RSPN\_exit\_from\_previous\_Clip. The stream data which were able to attach the shadow before RSPN\_arrival\_time\_discontinuity of Bridge-Clip contained in TS1, They are the stream data from the source packet of the beginning of Bridge-Clip to the source packet in front of the source packet referred to by RSPN\_arrival\_time\_discontinuity.

[0310]TS2 in drawing 96 comprises the stream data which were able to attach the shadow of Clip2 (Clip AV stream), and the stream data which were able to attach the shadow after RSPN\_arrival\_time\_discontinuity of Bridge-Clip. The stream data which were able to attach the shadow after RSPN\_arrival\_time\_discontinuity of Bridge-Clip contained in TS2, They are the stream data from the source packet referred to by RSPN\_arrival\_time\_discontinuity to the source packet of the last of Bridge-Clip. The stream data which were able to attach the shadow of Clip2 of TS2, From the source packet referred to by RSPN\_enter\_to\_current\_Clip. They are the stream data to the address of a stream required in order to decode the presentation unit corresponding to OUT\_time (illustrated by OUT\_time2 in drawing 96) of the present PlayItem.

[0311]Drawing 97 shows the relation between PlayItem preceded when not using Bridge-Clip, and the present PlayItem. In this case, the stream data which a player reads give a shadow and are shown. TS1 in drawing 97 comprises the stream data which were able to attach the shadow of Clip1 (Clip AV stream). The stream data which were able to attach the shadow of Clip1 of TS1, It begins from the address of a stream required in order to decode the presentation unit corresponding to IN\_time (illustrated by IN\_time1 in drawing 97) of PlayItem to precede, and is data to the source packet of the last of Clip1. TS2 in drawing 97 comprises the stream data which were able to attach the shadow of Clip2 (Clip AV stream).

[0312]The stream data which were able to attach the shadow of Clip2 of TS2, They are the stream data to the address of a stream required in order to begin from the source packet of the beginning of Clip2 and to decode the presentation unit corresponding to OUT\_time (illustrated by OUT\_time2 in drawing 97) of the present PlayItem.

[0313]In drawing 96 and drawing 97, TS1 and T2 are the streams which the source packet followed. Next, stream regulation of TS1 and TS2 and the connection conditions between them are considered. First, the coding restrictions for seamless connection are considered. As restriction of the coding structure of a transport stream, the number of the programs included in TS1 and TS2 must be 1 first. The number of the video streams contained in TS1 and TS2 must be 1. The number of the audio streams contained in TS1 and TS2 must be two or less. The number of the audio streams contained in TS1 and TS2 must be equal. In TS1 and/or TS2, elementary streams or private streams other than the above may be contained.

[0314]Restriction of a video bit stream is explained. Drawing 98 is a figure showing the example of the seamless connection shown by the display order of a picture. In order to be able to display a video stream seamlessly in a node, The unnecessary picture displayed the OUT\_time1 (OUT\_time of Clip1) back and before IN\_time2 (IN\_time of Clip2) must be removed by the process of re-encoding the partial stream of Clip near a node.

[0315]When shown in drawing 98, the example which makes seamless connection using BridgeSequence is shown in drawing 99. The video stream of Bridge-Clip before RSPN\_arrival\_time\_discontinuity comprises the coding video stream to the picture corresponding to OUT\_timeof Clip1 of drawing 98 1. And it is connected to the video stream of Clip1 to precede, and the video stream is re-encoded so that it may become the elementary stream which followed the MPEG 2 standard by one continuation.

[0316] Similarly, the video stream of Bridge-Clip after RSPN\_arrival\_time\_discontinuity comprises the coding video stream after the picture corresponding to IN\_timeof Clip2 of drawing 98 2. And a decoding start can be carried out correctly and it is connected to the video stream of Clip2 following this, and the video stream is re-encoded so that it may become the elementary stream which followed the MPEG 2 standard by one continuation. In order to make Bridge-Clip, generally, the picture of several sheets must be re-encoded and the other picture can be copied from original Clip.

[0317] The example which makes seamless connection without using BridgeSequence in the case of the example shown in drawing 98 is shown in drawing 100. The video stream of Clip1 comprises the coding video stream to the picture corresponding to OUT\_time1 of drawing 98, and it is re-encoded so that it may become the elementary stream which followed the MPEG 2 standard by one continuation. Similarly, the video stream of Clip2 comprises the coding video stream after the picture corresponding to IN\_timeof Clip2 of drawing 98 2, and it is re-encoded so that it may become the elementary stream which followed the MPEG 2 standard by one continuation.

[0318] The frame rate of the video stream of TS1 and TS2 must be equal to explaining coding restrictions of a video stream first. The termination of the video stream of TS1 must be carried out by sequence\_end\_code. The video stream of TS2 must be started by Sequence Header, GOP Header, and I-picture. The video stream of TS2 must be started by closed GOP.

[0319] The video presentation unit (a frame or the field) defined in a bit stream must be continuation on both sides of a node. There must not be any gap of a frame or the field in a node. In a node, the field sequence of a top ? bottom product must be continuation. In encoding which uses 3-2 PURUDAUN, "top\_field\_first" It reaches. In order to rewrite a "repeat\_first\_field" flag or to prevent generating of a field gap, it may be made to re-encode locally.

[0320] If the sampling frequency of the audio of TS1 and TS2 is not the same, it will not be explaining coding restrictions of an audio bit stream. If the encoding method (example . the MPEG1 layer 2, AC-3, SESF LPCM, AAC) of the audio of TS1 and TS2 is not the same, it will not become.

[0321] Next, the audio frame of the last of the audio stream of TS1 must contain the audio sample with display time equal at the time of the end of a display of the display picture of the last of TS1 in explaining coding restrictions of MPEG-2 transport stream. The audio frame of the beginning of the audio stream of TS2 must contain the audio sample with display time equal at the time of the display start of the display picture of the beginning of TS2.

[0322]In a node, the sequence of an audio presentation unit must not have a gap. As shown in Drawing 101, there may be overlap defined by the length of the audio presentation unit of less than 2 audio frame sections. The first packet that transmits the elementary stream of TS2 must be a video packet. The transport stream in a node must follow DVR-STD mentioned later.

[0323]TS1 and TS2 must not contain the break point of arrival time base in explaining restriction of Clip and Bridge-Clip in each.

[0324]The following restrictions are applied only when using Bridge-Clip. Only in the node of the source packet of the last of TS1, and the source packet of the beginning of TS2, a Bridge-ClipAV stream has a break point of only one arrival time base. RSPN\_arrival\_time\_discontinuity defined in ClipInfo() must show the address of the break point, and it must show the address which refers to the source packet of the beginning of TS2.

[0325]May any source packet in Clip1 be sufficient as the source packet referred to by RSPN\_exit\_from\_previous\_Clip defined in BridgeSequenceInfo()? It does not need to be a boundary of Aligned unit. May any source packet in Clip2 be sufficient as the source packet referred to by RSPN\_enter\_to\_current\_Clip defined in BridgeSequenceInfo()? It does not need to be a boundary of Aligned unit.

[0326]OUT\_time (OUT\_time1 shown in drawing 96 and drawing 97) of PlayItem preceded for explaining restriction of PlayItem must show the display finish time of the video presentation unit of the last of TS1. IN\_time (IN\_time2 shown in F drawing 96 and drawing 97) of the present PlayItem must show the display start time of the video presentation unit of the beginning of TS2.

[0327]Seamless connection must be made by explaining restriction of the data allocation in the case of using Bridge-Clip with reference to Drawing 102 so that the continuous supply of data may be guaranteed by a file system. This must be performed by arranging the Bridge-Clip AV stream connected to Clip1 (ClipAV stream file) and Clip2 (Clip AV stream file) so that data allocation regulation may be fulfilled.

[0328]The stream portion of Clip1 (Clip AV stream file) before RSPN\_exit\_from\_previous\_Clip as arranged to the continuation field more than half fragmentation, RSPN\_exit\_from\_previous\_Clip must be chosen. The data length of a Bridge-Clip AV stream must be chosen so that it may be arranged to the continuation field more than half fragmentation. The stream portion of Clip2 (Clip AV stream file) after RSPN\_enter\_to\_current\_Clip as arranged to the continuation field more than half fragmentation, RSPN\_enter\_to\_current\_Clip must be chosen.

[0329]Seamless connection must be made by explaining restriction of the data

allocation in the case of making seamless connection without using Bridge-Clip with reference to Drawing 103 so that the continuous supply of data may be guaranteed by a file system. This must be performed by arranging the portion of the last of Clip1 (Clip AV stream file), and the portion of the beginning of Clip2 (Clip AV stream file) so that data allocation regulation may be fulfilled.

[0330]The stream portion of the last of Clip1 (Clip AV stream file) must be arranged to the continuation field more than half fragmentation. The stream portion of the beginning of Clip2 (Clip AV stream file) must be arranged to the continuation field more than half fragmentation.

[0331]Next, DVR-STD is explained. DVR-STD is a conceptual model for modeling generation of a DVR MPEG2 transport stream, and decoding in the case of verification. DVR-STD is also a conceptual model for modeling generation of the AV stream referred to by two PlayItem(s) which were mentioned above, and by which seamless connection was made, and decoding in the case of verification.

[0332]A DVR-STD model is shown in Drawing 104. The DVR MPEG-2 transport-stream player model is contained in the model shown in Drawing 104 as a component. The transcription method of  $n$ ,  $TB_n$ ,  $MB_n$ ,  $EB_n$ ,  $TB_{sys}$ ,  $B_{sys}$ ,  $R_{xn}$ ,  $R_{bxn}$ ,  $R_{xsys}$ ,  $D_n$ ,  $D_{sys}$ ,  $O_n$ , and  $P_n(k)$  is the same as what is defined as T-STD of ISO/IEC 13818-1. That is, it is as follows.  $n$  is an index number of an elementary stream.  $TB_n$  is a transport buffer of the elementary stream  $n$ , and is \*\*.

[0333] $MB_n$  is a multiple buffer of the elementary stream  $n$ . It exists only about a video stream.  $EB_n$  is an elementary stream buffer of the elementary stream  $n$ . It exists only about a video stream.  $TB_{sys}$  is an input buffer for the system information of the program under decoding.  $B_{sys}$  is a main buffer in the system target decoder for the system information of the program under decoding.  $R_{xn}$  is a transmission rate by which data is removed from  $TB_n$ .  $R_{bxn}$  is a transmission rate by which a PES packet pay load is removed from  $MB_n$ . It exists only about a video stream.

[0334] $R_{xsys}$  is a transmission rate by which data is removed from  $TB_{sys}$ .  $D_n$  is a decoder of the elementary stream  $n$ .  $D_{sys}$  is a decoder about the system information of the program under decoding.  $O_n$  is re-ordering buffer of the video stream  $n$ .  $P_n(k)$  is the  $k$ -th presentation unit of the elementary stream  $n$ .

[0335]The decoding process of DVR-STD is explained. While reproducing DVR MPEG-2 single transport stream, the timing which inputs a transport packet into the buffer of  $TB_1$ ,  $TB_n$ , or  $TB_{sys}$  is determined by arrival\_time\_stamp of a source packet. Regulation of the buffering operation of  $TB_1$ ,  $MB_1$ ,  $EB_1$ ,  $TB_n$ ,  $B_n$ ,  $TB_{sys}$ , and  $B_{sys}$  is the same as T-STD specified to ISO/IEC 13818-1. Regulation of decoding operation

and a display action is the same as T-STD specified to ISO/IEC 13818-1.

[0336]The decoding process [ it is reproducing PlayItem by which seamless connection was made ] of a between is explained. Here, reproduction of two AV streams referred to by PlayItem by which seamless connection was made will be explained, and future explanation explains the reproduction of TS (for example, shown in drawing 96)1, and TS2 mentioned above. TS1 is a stream to precede and TS2 is the present stream.

[0337]Drawing 105 shows the timing chart of the input of a transport packet when moving from a certain AV stream (TS1) to the following AV stream (TS2) seamlessly connected to it, decoding, and a display. While moving from a predetermined AV stream (TS1) to the following AV stream (TS2) seamlessly connected to it, The time-axis (Drawing 105 is shown by ATC2) of the arrival time base of TS2 is not the same as that (Drawing 105 is shown by ATC1) of the arrival time base of TS1.

[0338]The time-axis (Drawing 105 is shown by STC2) of the system time base of TS2 is not the same as that (Drawing 105 is shown by STC1) of the system time base of TS1. It is required that the display of video should continue seamlessly. Overlap may be shown in the display time of the presentation unit of an audio.

[0339]The input timing to DVR-STD is explained. Until the video packet of the time to time  $T_1$ , i.e., the last of TS1, carries out the end of an input TB1 of DVR-STD, The input timing to the buffer of TB1 of DVR-STD, TBn, or TBsys is determined by arrival\_time\_stamp of the source packet of TS1.

[0340]The remaining packets of TS1 must be inputted into the buffer of TBn of DVR-STD, or TBsys by the bit rate of TS\_recording\_rate (TS1). Here, TS\_recording\_rate (TS1) is a value of TS\_recording\_rate defined in ClipInfo() corresponding to Clip1. The time which the byte of the last of TS1 inputs into a buffer is time  $T_2$ . Therefore, arrival\_time\_stamp of a source packet is disregarded in the section from time  $T_1$  to  $T_2$ .

[0341]When N1 is made into the number of bytes of the transport packet of TS1 following the video packet of the last of TS1, time DT1 to time  $T_1$  thru/or  $T_2$ , It is time required in order that N1 byte may carry out the end of an input by the bit rate of TS\_recording\_rate (TS1), and is computed by a following formula.

$DT1 = T_2 - T_1 = N1 / TS\_recording\_rate$  Both the values of RXn and RXsys change to the value of TS\_recording\_rate (TS1) before time (TS1)  $T_1$  thru/or  $T_2$ . Buffering operation other than this rule is the same as T-STD.

[0342]arrival time clock counter is reset by the value of arrival\_time\_stamp of the source packet of the beginning of TS2 in the time of  $T_2$ . The input timing to the buffer

of TB1 of DVR-STD, TBn, or TBsys is determined by arrival\_time\_stamp of the source packet of TS2. RXn and RXsys both change to the value defined in T-STD.

[0343]About additional audio buffering and system-data buffering to explain an audio decoder and a system decoder, In addition to the amount of buffers defined by T-STD, the additional amount of buffers (data volume for about 1 second) is required so that the input data of the section from the time T1 to T2 can be processed.

[0344]The display of a video presentation unit must let a node pass to explain the presentation timing of video, and it must be continuation without a gap. Here, STC1 considers it as the time-axis (in Drawing 105, illustrated with STC1) of the system time base of TS1, and STC2 is a time-axis (in drawing 97, illustrated with STC2.) of the system time base of TS2. Correctly, STC2 is started from the time which PCR of the beginning of TS2 inputted into T-STD. It carries out.

[0345]The offset between STC1 and STC2 is determined as follows.  $PTS_{end}^1$  is PTS on STC1 corresponding to the video presentation unit of the last of TS1, and  $PTS_{start}^2$  Are PTS on STC2 corresponding to the video presentation unit of the beginning of TS2, and  $T_{pp}$ . If it is a display period of the video presentation unit of the last of TS1, offset STC\_delta between two system time base will be computed by a following formula.

$STC\_delta = PTS_{end}^1 + T_{pp} - PTS_{start}^2$  [0346]In a node to explain the timing of the presentation of an audio, There may be overlap of the display timing of an audio presentation unit, and it is 0 thru/or less than 2 audio frames (see "audio overlap" currently illustrated by Drawing 105). Which audio sample being chosen and carrying out resynchronization of the display of an audio presentation unit to the amended time base after a node are set up by the player side.

[0347]In time  $T_5$ , the audio presentation unit of the last of TS1 is displayed for explaining about the system time clock of DVR-STD. The system time clock may overlap from time  $T_2$  to  $T_5$ . In this section, DVR-STD changes a system time clock between the value (STC1) of old time base, and the value (STC2) of new time base. The value of STC2 is computed by a following formula.

$STC2 = STC1 - STC\_delta$  [0348]The continuity of buffering is explained.  $STC1_{video\_end}^1$  is a value of STC on system time base STC1 in case the byte of the last of the video packet of the last of TS1 arrives to TB1 of DVR-STD.  $STC2_{video\_start}^2$  is a value of STC on system time base STC2 in case the byte of the beginning of the video packet of the beginning of TS2 arrives to TB1 of DVR-STD.  $STC2_{video\_end}^1$  is the value which converted the value of  $STC1_{video\_end}^1$  into the value on system time base STC2.  $STC2_{video\_end}^1$  is computed by a following formula.



$STC2_{video\_end}^1 = STC1_{video\_end}^1 - STC\_delta$  [0349] In order to follow DVR-STD, it is required that the following two conditions should be fulfilled. First, the arrival timing of TB1 of the video packet of the beginning of TS2 must fill the inequality shown below. And the inequality shown below must be filled.

It reaches Clip1 so that the inequality of  $STC2_{video\_start}^2 > STC2_{video\_end}^1 + \Delta T_1$  \*\* may be filled, or the partial stream of Clip2 -- re-encoding -- and -- or when it is necessary to re-multiplex-ize, it is carried out if needed [ the ].

[0350] next, the input of the video packet from TS2 which continues at the input of the video packet from TS1, and it on the time-axis of the system time base which converted STC1 and STC2 on the same time-axis -- a video buffer -- overflow -- and don't carry out underflow.

[0351] The contents of the data currently recorded on the recording medium by being based on such syntax, a data structure, and a rule, Reproduction information etc. can be managed appropriately and it has them, and a user can check the contents of the data currently appropriately recorded on the recording medium at the time of reproduction, or it can make it possible to reproduce desired data simple.

[0352] Although this embodiment makes an MPEG2 transport stream an example and explains it as a multiplexed stream, It is possible to apply also to the DSS transport stream currently used with DirecTV service (trademark) of not only this but an MPEG 2 program stream or the U.S.

[0353] Next, the syntax of mark\_entry() and representative\_picture\_entry(), The processing in the case of performing search reproduction of the scene shown with the marking point in the case of being composition as shown in drawing 81 is explained with reference to the flow chart of Drawing 106.

[0354] In Step S1, first the control section 23 of the recording and reproducing device 1, EP\_Map (drawing 70), STC\_Info (drawing 52), Program\_Info (drawing 54), and ClipMark (drawing 78) which are the data dace of a DVR transport stream file are read from the recording medium 100.

[0355] The control section 23 in Step S2 representative\_picture\_entry of ClipMark (drawing 78) (drawing 81), Or the list of thumbnails is created from the picture referred to by ref\_thumbnail\_index, and it outputs from the terminal 24 as user interface input and output, and is made to display on the menu screen of GUI. In this case, priority is given to ref\_thumbnail\_index over representative\_picture\_entry when ref\_thumbnail\_index has an effective value.

[0356] In Step S3, a user specifies the marking point of a reproducing starting point. This is performed because a user chooses a thumbnail image from the inside on the

menu screen displayed as GUI for example. The control section 23 acquires the marking point matched with the specified thumbnail corresponding to this selection operation.

[0357]In step S4, the control section 23 acquires STC\_sequence\_id with PTS of mark\_time\_stamp of mark\_entry (drawing 81) specified at Step S3.

[0358]In Step S5, the control section 23 acquires the source packet number which the STC time-axis corresponding to STC\_sequence\_id acquired by step S4 starts from STC\_Info (drawing 52).

[0359]In Step S6, in time the control section 23 from PTS of a marking point from PTS of the packet number which the STC time-axis acquired at Step S5 starts, and the marking point acquired by step S4 in front, And a source packet number with the nearest entry point (I picture) is acquired.

[0360]The control section 23 reads the data of a transport stream, and is made to supply it to AV decoder 27 in Step S7 from a source packet number with the entry point acquired at Step S6.

[0361]The control section 23 controls AV decoder 27, and makes a display start in Step S8 from the picture of PTS of the marking point acquired by step S4.

[0362]The above operation is further explained with reference to Drawings 107 thru/or 109.

[0363]Now, a DVR transport stream file has an STC time-axis of STC\_sequence\_id=id0, and let the source packet number which the time-axis starts be a thing smaller than the source packet number of the scene starting point A as shown in Drawing 107. And CM (commercials) shall be inserted in from the source packet number B before C.

[0364]At this time, to EP\_Map corresponding to EP\_Map shown in drawing 70. As shown in Drawing 108, corresponding to A, B, and C which are shown by RSPN\_EP\_start, each PTS is registered as PTS (A), PTS (B), and PTS (C) as PTS\_EP\_start.

[0365]As shown in Drawing 109, to ClipMark corresponding to ClipMark of drawing 78. As shown in Drawing 109, mark\_entry and representative\_picture\_entry are recorded corresponding to the value of a scene start, CM start and the mark type (drawing 79) 0x92 showing CM end, 0x94, and 0x95.

[0366]As Mark\_Time\_stamp of mark\_entry, Corresponding to the scene start, CM start, and CM end, PTS (a1), PTS (b0), and PTS (c0) are registered, respectively, and each of each STC\_sequence\_id is set to id0.

[0367]Similarly as Mark\_Time\_stamp of Representative\_picture\_entry, Corresponding

to the scene start, CM start, and CM end, PTS (a2), PTS (b0), and PTS (c0) are registered, respectively, and, as for each of them, STC\_sequence\_id is set to id0.

[0368]In [ in PTS(A) < PTS (a1), packet number A is acquired in Step S6, and ] Step S7, The transport stream which begins from packet number A is supplied to AV decoder 27, and a display is started from the picture of PTS (a1) in Step S8.

[0369]Next, with reference to the flow chart of Drawing 110 the syntax of mark\_entry and representative\_picture\_entry, Processing of the CM skip reproduction in the case of being composition as shown in drawing 81 is explained with reference to the flow chart of Drawing 110.

[0370]In Step S21, the control section 23 reads EP\_map (drawing 70), STC\_Info (drawing 52), Program\_Info (drawing 54), and ClipMark (drawing 78) from the recording medium 100. In Step S22, a user specifies CM skip reproduction from the terminal 24 as user interface input and output.

[0371]In Step S23, the control section 23 acquires STC\_sequence\_id to which the mark information which is PTS and CM end point (0x95) of the mark information whose mark type (drawing 79) is CM starting point (0x94) PTS(s) and corresponds (drawing 81).

[0372]In Step S24, the control section 23 acquires the source packet number which the STC time-axis corresponding to STC\_sequence\_id of CM starting point and an end point starts from STC\_Info (drawing 52).

[0373]The control section 23 makes a transport stream read from the recording medium 100, supplies it to AV decoder 27, and makes decoding start in Step S25.

[0374]In Step S26, the control section 23 investigates whether the present display image is a picture of PTS of CM starting point. When the present display image is not a picture of PTS of CM starting point, it progresses to Step S27 and, as for the control section 23, the display of a picture is continued. Then, processing returns to Step S25 and repeat execution of the processing after it is carried out.

[0375]In Step S26, when judged with the present display image being a picture of PTS of CM starting point, it progresses to Step S28, and the control section 23 controls AV decoder 27, and stops decoding and a display.

[0376]Next, in Step S29 the control section 23, Acquire the packet number which the STC time-axis corresponding to STC\_sequence\_id of CM end point starts, and The packet number, Before being more nearly time than PTS of the point, a source packet number with the nearest entry point is acquired from PTS of CM end point acquired by processing of Step S23.

[0377]The control section 23 reads the data of a transport stream, and is made to

supply it to AV decoder 27 in Step S30 from a source packet number with the entry point acquired by processing of Step S29.

[0378]The control section 23 controls AV decoder 27, and makes a display resume from the picture of PTS of CM end point in Step S31.

[0379]When the above operation is further explained with reference to Drawing 107 thru/or Drawing 109, CM starting point and CM end point, In the case of this example, it exists on a common STC time-axis called STC\_sequence\_id=id0, and let the source packet number which that STC time-axis starts be a thing smaller than the source packet number A of the starting point of a scene.

[0380]A transport stream is decoded, and a display is suspended by AV decoder 27 when judged with display time having been set to PTS (b0) at Step S26 (when judged with it being CM starting point). And in PTS(C) < PTS (c0), at Step S30, decoding is resumed from the stream which begins from the data of packet number C, and a display is resumed from the picture of PTS (c0) in Step S31.

[0381]This method can be applied, not only CM skip reproduction but when skipping the scene for two points generally specified by ClipMark and reproducing.

[0382]Next, mark\_entry and representative\_picture\_entry explain search regeneration of CM shown with the marking point in the case of being the syntax structure shown in drawing 82 with reference to the flow chart of Drawing 112.

[0383]In Step S41, the control section 23 acquires the information on EP\_map (drawing 70), STC\_Info (drawing 52), Program\_Info (drawing 54), and ClipMark (drawing 78).

[0384]Next, in Step S42 the control section 23, The list of thumbnails is generated from the picture referred to by representative\_picture\_entry (drawing 82) or ref\_thumbnail\_index contained in ClipMark (drawing 78) read at Step S41, It is made to display on the menu screen of GUI. Priority is given to ref\_thumbnail\_index over representative\_picture\_entry when ref\_thumbnail\_index has an effective value.

[0385]In Step S43, a user specifies the marking point of a reproducing starting point. This specification is performed by specifying the marking point in which a user chooses a thumbnail image, is matched with that thumbnail at, and is, for example from the inside on the menu screen displayed by processing of Step S42.

[0386]In Step S44, the control section 23 acquires RSPN\_ref\_EP\_start and offset\_num\_pictures (drawing 82) of a marking point which were specified by processing of Step S43.

[0387]The control section 23 reads the data of a transport stream from the source packet number corresponding to RSPN\_ref\_EP\_start acquired at Step S44, and is

made to supply it to AV decoder 27 in Step S45.

[0388]In Step S46, the control section 23 controls AV decoder 27, When the picture which should be displayed (without) is counted up from the picture referred to by RSPN\_ref\_EP\_start and counted value becomes offset\_num\_pictures, a display is made to start from the picture.

[0389]The above processing is further explained with reference to Drawing 113 thru/or Drawing 115. In this example, the scene has started the DVR transport stream file from the source packet number A, and CM is inserted from the source packet number B to the source packet C. For this reason, as shown in Drawing 114, corresponding to A as RSPN\_EP\_start, B, and C, PTS (A), PTS (B), and PTS (C) are registered into EP\_map as PTS\_EP\_start.

[0390]As shown in Drawing 115, mark\_entry and representative\_picture\_entry are registered corresponding to the scene start, CM start, and the mark type of CM end. Corresponding to the scene start, CM start, and CM end, as RSPN\_ref\_EP\_start, A, B, and C are registered into mark\_entry, respectively, and M1, N1, and N2 are registered into it as offset\_num\_pictures. Similarly to representative\_picture\_entry. As RSPN\_ref\_EP\_start, corresponding to the scene start, CM start, and CM end, A, B, and C are registered, respectively and M2, N1, and N2 are registered as offset\_num\_pictures, respectively.

[0391]When it \*\*\*\*(ed) from the picture which is in charge of a scene start and is ordered in reproduction, Decoding is started from the stream which begins from the data of packet number A, When the picture which should be displayed from the picture of PTS (A) (without it displays) is counted up and offset\_num\_pictures becomes a value of M1, a display is started from the picture.

[0392]The syntax of mark\_entry and representative\_picture\_entry explains processing of the CM skip reproduction in the case of being the composition shown in drawing 82 with reference to the flow chart of Drawing 116.

[0393]In Step S61, the control section 23 acquires the information on EP\_map (drawing 70), STC\_Info (drawing 52), Program\_Info (drawing 54), and ClipMark (drawing 78).

[0394]In Step S62, when a user orders it CM skip reproduction, in Step S63 the control section 23, As mark information of each point which is CM starting point and a CM end point, a mark type (drawing 79) acquires RSPN\_ref\_EP\_START and offset\_num\_pictures (drawing 82). And the data of CM starting point is made into RSPN\_ref\_EP\_start (1) and offset\_num\_pictures (1), and the data of CM end point is made into RSPN\_ref\_EP\_start (2) and offset\_num\_pictures (2).

[0395]In Step S64, the control section 23 acquires PTS corresponding to RSPN\_ref\_EP\_start (1) and RSPN\_ref\_EP\_start (2) from EP\_map (drawing 70).

[0396]The control section 23 makes a transport stream read from the recording medium 100, and is made to supply to AV decoder 27 in Step S65.

[0397]In Step S66, the control section 23 judges whether the present display image is a picture of PTS corresponding to RSPN\_ref\_EP\_start (1). When the present display image is not a picture of PTS corresponding to RSPN\_ref\_EP\_start (1), it progresses to Step S67 and a picture is displayed continuously as it is. Then, processing returns to Step S65 and repeat execution of the processing after it is carried out.

[0398]When judged with the present display image being a picture of PTS corresponding to RSPN\_ref\_EP\_start (1) in Step S66, Progress to Step S68 and the control section 23 controls AV decoder 27, A display is stopped, when the picture displayed from the picture of PTS corresponding to RSPN\_ref\_EP\_start (1) is counted up and counted value becomes offset\_num\_pictures (1).

[0399]The control section 23 reads the data of a transport stream from the source packet number of RSPN\_ref\_EP\_start (2), and is made to supply it to AV decoder 27 in Step S69.

[0400]In Step S70, the control section 23 controls AV decoder 27, When the picture which should be displayed from the picture of PTS corresponding to RSPN\_ref\_EP\_start (2) (without it displays) is counted up and counted value becomes offset\_num\_pictures (2), a display is made to start from the picture.

[0401]If the above operation is further explained with reference to Drawing 113 thru/or Drawing 115, the time PTS corresponding to packet number B and C (B) and PTS (C) will be first obtained based on EP\_map (Drawing 114). And when Clip AV stream is decoded and display time is set to PTS (B), a display picture counts up from the picture of PTS (B), and a display is suspended when the value is set to N1 (Drawing 115).

[0402]Decoding is resumed from the stream which begins from the data of packet number C, and when the picture which should be displayed from the picture of PTS (C) (without it displays) is counted up and the value is set to N2 (Drawing 115), a display is resumed from the picture.

[0403]The above processing can be applied, not only CM skip reproduction but when making the scene for two points specified by ClipMark skip and reproducing.

[0404]Next, the syntax of mark\_entry and representative\_picture\_entry explains search regeneration of the scene shown with the marking point in the case of being composition as shown in drawing 84 with reference to the flow chart of Drawing 118.

[0405]In Step S81, the information on EP\_map (drawing 70), STC\_Info (drawing 52), Program\_Info (drawing 54), and ClipMark (drawing 78) is acquired.

[0406]In Step S82, the control section 23 generates the list of thumbnails from the picture referred to by representative\_picture\_entry or ref\_thumbnail\_index of ClipMark (drawing 78). It is made to display as a menu screen of GUI. Priority is given to ref\_thumbnail\_index over representative\_picture\_entry when ref\_thumbnail\_index has an effective value.

[0407]In Step S83, a user specifies the marking point of a reproducing starting point. A user chooses a thumbnail image from the inside on a menu screen, and this specification is performed by specifying the marking point matched with that thumbnail, for example.

[0408]In Step S84, the control section 23 acquires RSPN\_mark\_point (drawing 84) of mark\_entry specified by the user.

[0409]In Step S85, the control section 23 is before RSPN\_mark\_point of a marking point, and acquires the nearest source packet number of an entry point from EP\_map (drawing 70).

[0410]The control section 23 reads the data of a transport stream from the source packet number corresponding to the entry point acquired at Step S85, and is made to supply it to AV decoder 27 in Step S86.

[0411]The control section 23 controls AV decoder 27, and makes a display start in Step S87 from the picture referred to by RSPN\_mark\_point.

[0412]The above processing is further explained with reference to Drawing 119 thru/or Drawing 121. In this example, a DVR transport stream file carries out a scene start by the source packet A, and CM is inserted from the source packet number B to C. For this reason, corresponding to A as RSPN\_EP\_start, B, and C, PTS\_EP\_start is registered into EP\_map of Drawing 120 as PTS (A), PTS (B), and PTS (C), respectively. It corresponds to ClipMark shown in Drawing 121 at a scene start, CM start, and CM end, a1, b1, and c1 are registered as RSPN\_mark\_point of markentry, and a2, b1, and c1 are registered as RSPN\_mark\_point of representative\_picture\_entry, respectively.

[0413]When **\*\*\***(ing) and reproducing from the picture which is in charge of a scene start, if packet number  $A < a1$ , decoding will be started from the stream which begins from the data of packet number A, and a display will be started from the picture corresponding to the source packet number a1.

[0414]Next, the syntax of mark\_entry and representative\_picture\_entry explains processing of the CM skip reproduction in the case of being composition as shown in drawing 84 with reference to the flow chart of Drawing 122 and Drawing 123.

[0415]In Step S101, the control section 23 acquires the information on EP\_map (drawing 70), STC\_Info (drawing 52), Program\_Info (drawing 54), and ClipMark (drawing 70).

[0416]In Step S102, a user specifies CM skip reproduction.

[0417]In Step S103, the control section 23 acquires RSPN\_mark\_point (drawing 84) of the mark information of each point whose mark types (drawing 79) are CM starting point and CM end point. And the control section 23 makes the data of CM starting point RSPN\_mark\_point (1), and makes the data of CM end point RSPN\_mark\_point (2).

[0418]The control section 23 makes a transport stream read from the recording medium 100, and AV decoder 27 is made to output and decode it in Step S104.

[0419]In Step S105, the control section 23 judges whether the present display image is a picture corresponding to RSPN\_mark\_point (1). When the present display image is not a picture corresponding to RSPN\_mark\_point (1), it progresses to Step S106 and a picture is displayed continuously as it is. Then, processing returns to Step S104 and repeat execution of the processing after it is carried out.

[0420]In Step S105, when it judges that the present display image is a picture corresponding to RSPN\_mark\_point (1), it progresses to Step S107, and the control section 23 controls AV decoder 27, and stops decoding and a display.

[0421]Next, in Step S108, the source packet number which is before RSPN\_mark\_point (2) and has the nearest entry point is acquired from EP\_map (drawing 70).

[0422]The control section 23 reads the data of a transport stream from the source packet number corresponding to the entry point acquired at Step S108, and is made to supply it to AV decoder 27 in Step S109.

[0423]The control section 23 controls AV decoder 27, and makes a display resume in Step S110 from the picture referred to by RSPN\_mark\_point (2).

[0424]A display is suspended, when the example of Drawing 119 thru/or Drawing 121 explained the above processing further, and Clip AVstream is decoded, it goes and it becomes a display picture corresponding to the source packet number b1 (Drawing 121). And when it was the source packet number  $C < \text{source packet number } c1$ , and decoding is resumed and it becomes a picture corresponding to the source packet number c1 from the stream which begins from the data of packet number C, a display is resumed from the picture.

[0425]As shown in Drawing 124 as mentioned above, on PlayList, A position can be specified with a time stamp, this time stamp can be changed into a data address in Clip Information of each Clip, and the position of Clip AV stream can be accessed.



[0426]In [ as more specifically shown in Drawing 125 ] a PlayList top, If a user specifies a bookmark and resume points as a time stamp on a time-axis as PlayListMark, when reproducing the PlayList, ClipMark of Clip which the PlayList is referring to can be used, and the scene starting point and the scene end point of Clip AV stream can be accessed.

[0427]The syntax of ClipMark is changed to the example of drawing 78, and can be shown in Drawing 126.

[0428]RSPN\_mark is changed and inserted in reserved\_for\_MakerID, mark\_entry(), and repretative\_picture\_entry() of drawing 78 in this example. The 32-bit field of this RSPN\_mark shows the relative address of the source packet which contains the 1st byte that is an access unit which that mark refers to on an AV stream file. RSPN\_mark is a size which makes a source packet number a unit, in Clip Information file, is defined from the source packet of the beginning of an AV stream file, and counts the value of offset\_SPN as an initial value.

[0429]Other composition is the same as that of the case in drawing 78.

[0430]The syntax of ClipMark can also be constituted as further shown in Drawing 127. In this example, RSPN\_ref\_EP\_start and offset\_num\_pictures are inserted instead of RSPN\_mark in Drawing 126. These are the same as that of the case where it is shown in drawing 82.

[0431]Next, Drawing 128 shows example of another of the syntax of ClipInfo().

[0432]Clip\_service\_type shows the type of an AV stream file. For example, Clip\_service\_type shows the type of video recording, audio recording, etc. For example, Clip\_service\_type may give the same meaning as the service type which the program of digital TV broadcasting shows. For example, in the case of digital BS broadcasting of Japan, a service type has three kinds, television services, voice service, and data-broadcasting service. The value representing the service type of the program which an AV stream includes is set to Clip\_service\_type.

[0433]MPEG 2 by which transcode\_mode\_flag was received from digital broadcasting It is a flag which shows the record method of a transport stream. When this flag is set to 1, it is shown that recoding of at least one elementary stream in the AV stream file corresponding to Clip was carried out, and it was recorded. When this flag is set to 1, it is shown that all the elementary streams in the AV stream file corresponding to Clip were recorded without changing anything by the contents received from digital broadcasting.

[0434]The other syntax fields have the same meaning as the field of the same name explained by drawing 46.

[0435]Next, example of another of ProgramInfo() is explained with reference to Drawing 129.

[0436]The contents of a program which this format specifies in an AV stream file call a fixed source packet sequence program-sequence.

[0437]The address which new program-sequence starts in an AV stream file is stored in ProgramInfo(). This address is shown by SPN\_program\_sequence\_start.

[0438]program-sequence other than program-sequence of the last in an AV stream file, It starts from the source packet to which it is pointed out by the SPN\_program\_sequence\_start, and ends by the source packet in front of the source packet to which it is pointed out by the following SPN\_program\_sequence\_start. It starts from the source packet to which it is pointed out by the SPN\_program\_sequence\_start, and the last program-sequence is ended by the source packet of the last of an AV stream file.

[0439]program-sequence may also straddle the boundary of STC-sequence.

[0440]length shows the number of bytes from the byte just behind this length field to the byte of the last of ProgramInfo().

[0441]num\_of\_program\_sequences shows the number of program-sequence in an AV stream file.

[0442]SPN\_program\_sequence\_start shows the address which program-sequence starts on an AV stream file. SPN\_program\_sequence\_start is a size which makes a source packet number a unit, and counts zero as an initial value from the source packet of the beginning of an AV stream file.

[0443]The value of SPN\_program\_sequence\_start entered in ProgramInfo() is located in a line with the ascending order.

[0444]SPN\_program\_sequence\_start is premised on having pointed out the source packet with the first PMT to the program-sequence. SPN\_program\_sequence\_start is made when a record machine (recording and reproducing device 1 of drawing 1) analyzes PSI/SI in a transport stream. Since a time delay until a record machine analyzes PSI/SI and detects the change is required, SPN\_program\_sequence\_start may point out the source packet which is within predetermined time from the change point of actual PSI/SI.

[0445]program\_map\_PID is a value with PMT (program map table) applicable to the program-sequence of PID of a transport packet.

[0446]num\_of\_streams\_in\_ps shows the number of the elementary streams defined in the program-sequence.

[0447]num\_of\_groups shows the number of the groups of the elementary stream

defined in the program-sequence. num\_of\_groups is one or more values.

[0448]When PSI/SI of a transport stream has the group information of an elementary stream, num\_of\_groups assumes taking one or more values. Each group constitutes one view in a multi view program.

[0449]stream\_PID shows the value of PID to the elementary stream defined in PMT which program\_map\_PID of the program-sequence refers to.

[0450]StreamCodingInfo() shows the information on the elementary stream to which it is pointed out by above-mentioned stream\_PID. It mentions later for details.

[0451]num\_of\_streams\_in\_group shows the number of elementary streams which the group of an elementary stream has.

[0452]stream\_index shows the value of stream\_index defined by the turn described by for-loop of stream\_index in the syntax corresponding to the elementary stream which the group of the above-mentioned elementary stream has.

[0453]Drawing 30 shows the syntax of StreamCodingInfo().

[0454]length shows the number of bytes from the byte just behind this length field to the byte of the last of StreamCodingInfo().

[0455]stream\_coding\_type shows the coding type of the elementary stream to which it is pointed out by stream\_PID corresponding to this StreamCodingInfo(). The meaning of a value is shown in Drawing 131.

[0456]video\_format shows the format video of the video stream to which it is pointed out by stream\_PID corresponding to this StreamCodingInfo().

[0457]The meaning of a value is the same as drawing 56.

[0458]frame\_rate shows the frame rate of the video stream to which it is pointed out by stream\_PID corresponding to this StreamCodingInfo().

[0459]The meaning of a value is the same as drawing 57.

[0460]display\_aspect\_ratio shows the display aspect ratio of the video stream to which it is pointed out by stream\_PID corresponding to this StreamCodingInfo().

[0461]The meaning of a value is the same as drawing 58.

[0462]cc\_flag is a flag which shows whether the closed caption (closed caption data) signal is coded in the video stream to which it is pointed out by stream\_PID corresponding to this StreamCodingInfo().

[0463]original\_video\_format\_flag is a flag which shows whether original\_video\_format and original\_display\_aspect\_ratio exist in this StreamCodingInfo().

[0464]original\_video\_format is the original format video before the video stream to which it is pointed out by stream\_PID corresponding to this StreamCodingInfo() is coded. The meaning of a value is the same as above-mentioned video\_format (drawing

56).

[0465]original\_display\_aspect\_ratio is the original display aspect ratio before the video stream to which it is pointed out by stream\_PID corresponding to this StreamCodingInfo() is coded.The meaning of a value is the same as above-mentioned display\_aspect\_ratio (drawing 58).

[0466]In a transformer and the case where it codes the transport stream which multimedia data streams (a BML stream, a title, etc.) have multiplexed with the video stream,By re-encoding a video stream, the format video changes (for example, it changes to 480i from 1080i), and, on the other hand, a multimedia data stream considers the case where the contents with an original stream are maintained. At this time, the mismatch of information may arise between a new video stream and a multimedia data stream. For example, in spite of deciding the parameter about the display of a multimedia data stream supposing the format video of an original video stream, it is a case where the format video changes with re-encodings of a video stream. In such a case, the information about an original video stream is saved at original\_video\_format and original\_display\_aspect\_ratio. A reproduction machine (recording and reproducing device 1 of drawing 1) builds a display image with an above-mentioned new video stream and multimedia data stream as follows.

[0467]- A video stream is raised and sampled by the format video shown by original\_video\_format and original\_display\_aspect\_ratio.- its rise, and the picture and multimedia data stream which were sampled are compounded, and build a right display image.

[0468]audio\_presentation\_type shows the presentation type of the audio stream to which it is pointed out by stream\_PID corresponding to this StreamCodingInfo().

[0469]The meaning of a value is the same as audio\_component\_type of drawing 61.

[0470]sampling\_frequency shows the sampling frequency of the audio stream to which it is pointed out by stream\_PID corresponding to this StreamCodingInfo().

[0471]The meaning of a value is the same as drawing 62.

[0472]Next, example of another of EP\_map is shown. This EP\_map does not store the data of PTS\_EP\_start and RSPN\_EP\_start explained by drawing 70 and drawing 72 in EP\_map as it is, but compression encoding is carried out for reduction of data volume, and it stores it in EP\_map (the control section 23 of drawing 1 performs).

[0473]EP\_map comprises the subday bull called at least one or more EP\_map\_for\_one\_stream\_PID(). Subday bull EP\_map\_for\_one\_stream\_PID() is made to one elementary stream transmitted by the transport packet with the value of the same PID.

[0474]EP\_map has the next information for all the EP\_map\_for\_one\_stream\_PID().

[0475](1) stream\_PID : specify PID of the transport packet which transmits the elementary stream to which it is pointed out by the EP\_map\_for\_one\_stream\_PID().

(2) EP\_stream\_type : the type of the entry point of the EP\_map\_for\_one\_stream\_PID() is shown.

(3) The entry of EP\_map\_for\_one\_stream\_PID(). : in which each entry has the following information –PTS\_EP\_start: The value of PTS of the access unit in an entry point is shown.

– RSPN\_EP\_start: The address which the access unit referred to by PTS\_EP\_start of the entry point starts in an AV stream is shown.

[0476]When EP\_map points out one or more elementary streams, EP\_map may have one or more EP\_map\_for\_one\_stream\_PID() sub tables. Drawing 67 shows the example of EP\_map when EP\_map has pointed out three video streams in Clip.

[0477]One EP\_map\_for\_one\_stream\_PID() is made by one table regardless of the break point of STC.

[0478]By comparing the value of RSPN\_STC\_start defined in the value of RSPN\_EP\_start and STC\_Info() shows the boundary of STC\_sequence in EP\_map (refer to drawing 68).

[0479]EP\_map has one EP\_map\_for\_one\_stream\_PID to the range which the stream transmitted by the same PID followed.

[0480]In the case of drawing 69, program#1 and program#3 have the same video PID, but since the data range is not continuing, it must have EP\_map\_for\_one\_stream\_PID for every program.

[0481]In order to reduce the data size of the table of EP\_map\_for\_one\_stream\_PID() and to improve the performance of data search, EP\_map\_for\_one\_stream\_PID() is divided into two sub tables, i.e., EP\_coarse and EP\_fine, (refer to Drawing 132).

[0482]An EP\_fine entry has the bit information by the side of LSB (Least Significant Bit) of PTS\_EP\_start and RSPN\_EP\_start (see Drawing 133 and Drawing 134). An EP\_coarse entry, PTS\_EP\_start. And the entry number in the bit information by the side of MSB (Most Significant Bit) of RSPN\_EP\_start, and the table of EP\_fine corresponding to it (the bit by the side of LSB taken out from the same PTS\_EP\_start.) It has an entry in the EP\_fine table which it has.

[0483]The number of entries in an EP\_coarse sub table is quite small compared with the number of entries of an EP\_fine sub table. In the following case. : from which the entry of EP\_coarse is made. – The entry of RSPN\_EP\_fine of the beginning after the value of the entry and RSPN\_EP\_fine of PTS\_EP\_fine of the beginning after the value

of the entry and PTS\_EP\_fine of the first PTS\_EP\_fine carries out a wrap around carries out a wrap around. (Refer to Drawing 135).

[0484]The example of the random access to the AV stream in the case of using EP\_map is explained.

[0485]Suppose that he would like to start the reproduction after 30 minutes on the global time-axis of a certain PlayList.

- Investigate STC-sequence-id of PlayItem which contains time the place which is on the global time-axis in PlayList first, and corresponds in 30 minutes.
- Draw the value of PTS corresponding to after 30 minutes on the local time-axis of aforementioned STC-sequence.
- Draw RSPN\_STC\_start of aforementioned STC-sequence from STC\_Info.
- In an EP\_coarse sub table, data search is started from the entry whose RSPN\_EP\_coarse is more than said RSPN\_STC\_start. In an EP\_coarse sub table, the entry of PTS\_EP\_coarse which has a front value in said required PTS in time than most the neighborhood and its value is found.
- In an EP\_fine sub table, data search is started from the entry number of EP\_fine corresponding to said found PTS\_EP\_coarse. An entry with PTS\_EP\_fine which has a front value in said required PTS in time than most the neighborhood and its value is found.
- RSPN\_EP\_start for the read-out start of the access unit of said required PTS, It is determined by replacing 18 bits of LSB of RSPN\_EP\_coarse corresponding to said found PTS\_EP\_coarse in the bit of RSPN\_EP\_fine corresponding to said found PTS\_EP\_fine.

[0486]The syntax of EP\_map which explained [ above-mentioned ] is shown in Drawing 136.

[0487]number\_of\_stream\_PID\_entries shows the number of entries of EP\_map\_for\_one\_stream\_PID in the inside of EP\_map.

[0488]stream\_PID [k] shows the value of PID of the transport packet which transmits the elementary stream referred to by EP\_map\_for\_one\_stream\_PID entered by the k-th in EP\_map.

[0489]EP\_stream\_type [k] The type of the elementary stream referred to by that of said EP\_map\_for\_one\_stream\_PID is shown. The meaning of a value is shown in the table of Drawing 137.

[0490]When EP\_stream\_type [k] is 0 or 1, the elementary stream is a video stream. The meaning of Video type 1 and video type 2 is mentioned later in the place of the explanation which is EP\_video\_type (Drawing 139).

[0491]When EP\_stream\_type [k] is 2, the elementary stream is an audio stream.

[0492]num\_EP\_coarse\_entries [k] The number of the EP-coarse entries in said EP\_map\_for\_one\_stream\_PID is shown.

[0493]num\_EP\_fine\_entries [k] The number of the EP-fine entries in said EP\_map\_for\_one\_stream\_PID is shown.

[0494]EP\_map\_for\_one\_stream\_PID\_start\_address [k] shows the relative byte position from which said EP\_map\_for\_one\_stream\_PID begins in EP\_map(). This value is shown by the number of bytes from the 1st byte of EP\_map().

[0495]The syntax of EP\_map\_for\_one\_stream\_PID is shown in Drawing 138. In order to explain the semantics of this, the meaning of PTS\_EP\_start which becomes the origin of the data stored in EP\_map\_for\_one\_stream\_PID, and RSPN\_EP\_start is explained.

[0496]PTS\_EP\_start and RSPN\_EP\_start related with it point out the entry point on an AV stream. And PTS\_EP\_fine and PTS\_EP\_coarse related with it are drawn from the same PTS\_EP\_start. RSPN\_EP\_fine and RSPN\_EP\_coarse related with it are drawn from the same RSPN\_EP\_start. PTS\_EP\_start and RSPN\_EP\_start are defined as follows.

[0497]PTS\_EP\_start is a unsigned integer of 33-bit length. The definition of PTS\_EP\_start changes with values of EP\_stream\_type to EP\_map\_for\_one\_stream\_PID.

[0498]When EP\_stream\_type is zero ('video type1'), PTS\_EP\_start shows PTS of 33 bit length of the video access unit started from a sequence header in an AV stream.

[0499]When EP\_stream\_type is 2 ('audio'), PTS\_EP\_start shows PTS of 33 bit length of the video access unit started from a sequence header in an AV stream.

[0500]When EP\_stream\_type is 1 ('video type2'), PTS\_EP\_start shows PTS of 33 bit length of the video access unit defined as Drawing 139 according to the value of EP\_video\_type related with it.

[0501]RSPN\_EP\_start is a 32-bit unsigned integer. The definition of RSPN\_EP\_start changes with values of EP\_stream\_type to EP\_map\_for\_one\_stream\_PID.

[0502]When EP\_stream\_type is zero ('video type1'), this field shows the address in the inside of the AV stream of the source packet containing the 1st byte of the video access unit related with PTS\_EP\_start.

[0503]When EP\_stream\_type is 2 ('audio'), this field shows the address in the inside of the AV stream of the source packet containing the 1st byte of the audio access unit related with PTS\_EP\_start.

[0504]When EP\_stream\_type is 1 ('video type2'), the meaning of RSPN\_EP\_start is defined as Drawing 139 according to the value of EP\_video\_type related with it.

[0505]RSPN\_EP\_start is expressed with the unit of a source packet number, and it

counts zero as an initial value from the source packet of the beginning in an AV stream file.

[0506]The semantics of EP\_map\_for\_one\_stream\_PID is explained.

[0507]EP\_fine\_table\_start\_address, The 1st byte of start address of the first EP\_video\_type [EP\_fine\_id] is shown, and it is expressed with the relative number of bytes from the 1st byte of EP\_map\_for\_one\_stream\_PID(). A relative number of bytes is started from zero.

[0508]ref\_to\_EP\_fine\_id shows the number with PTS\_EP\_fine related with PTS\_EP\_coarse following this field of an EP\_fine entry. PTS\_EP\_fine and PTS\_EP\_coarse related with it are drawn from the same PTS\_EP\_start.

[0509]ref\_to\_EP\_fine\_id is given with the value of EP\_fine\_id defined by the turn described in for-loop of EP\_fine\_id.

[0510]PTS\_EP\_coarse, PTS\_EP\_fine, and RSPN\_EP\_coarse and RSPN\_EP\_fine are drawn as follows.

[0511]A Nf piece entry is shown in an EP\_fine sub table, and these entries assume that it has ranked with the ascending order of the value of RSPN\_EP\_start relevant to them.

[0512]Each PTS\_EP\_fine entry is drawn as follows from the PTS\_EP\_start.

[0513]PTS\_EP\_fine[EP\_fine\_id] = % (PTS\_EP\_start [EP\_fine\_id] >> 9) 2<sup>12</sup>[0514]The relation between PTS\_EP\_coarse and its PTS\_EP\_fine is as follows.

[0515]PTS\_EP\_coarse[i] -- = % (PTS\_EP\_start [ref\_to\_EP\_fine\_id[i]] >> 19) 2

<sup>14</sup>TS\_EP\_fine [ref\_to\_EP\_fine\_id[i]]. = % (PTS\_EP\_start[ref\_to\_EP\_fine\_id[i]] >> 9) 2

<sup>12</sup>[0516]Each RSPN\_EP\_fine entry is drawn as follows from the RSPN\_EP\_start.

[0517]RSPN\_EP\_fine[EP\_fine\_id] = RSPN\_EP\_start[EP\_fine\_id] % 2<sup>18</sup>[0518]The relation between RSPN\_EP\_coarse and its RSPN\_EP\_fine is as follows.

[0519]RSPN\_EP\_coarse[i] =

RSPN\_EP\_start[ref\_to\_EP\_fine\_id[i]]RSPN\_EP\_fine[ref\_to\_EP\_fine\_id[i]]

=RSPN\_EP\_start[ref\_to\_EP\_fine\_id. [i]] % 2<sup>18</sup>[0520]EP\_video\_type [EP\_fine\_id], When EP\_stream\_type is 1 ('video type2'), the type of the video access unit which PTS\_EP\_fine [EP\_fine\_id] and RSPN\_EP\_fine [EP\_fine\_id] point out is shown. The meaning of a value is as being shown in Drawing 139.

[0521]EP\_map\_for\_one\_stream\_PID() fills the next restriction.

[0522]– The entry of RSPN\_EP\_coarse must be located in a line in ascending order of the value of RSPN\_EP\_coarse in EP\_map\_for\_one\_stream\_PID().

– The entry of RSPN\_EP\_fine must be located in a line in ascending order of the value of RSPN\_EP\_start relevant to it in EP\_map\_for\_one\_stream\_PID().



- In the following case. PTS\_EP\_coarse. : from which the entry of RSPN\_EP\_coarse is made. .. Entry of the first PTS\_EP\_fine .. Entry of PTS\_EP\_fine of the beginning after the value of PTS\_EP\_fine carries out a wrap around .. Entry of RSPN\_EP\_fine of the beginning after the value of RSPN\_EP\_fine carries out a wrap around.

[0523]Next, Drawing 140 shows the flow chart of the recording operation of a Clip AV stream file and the Clip Information file relevant to it. It explains with reference to the recording and reproducing device of drawing 1.

[0524]The transport stream obtained at Step S201 by the control section 23 encoding AV input inputted from the terminals 11 and 12, Or the transport stream inputted from the digital interface of the terminal 13 is file-ized, and a Clip AV stream file is created and recorded.

[0525]At Step S202, the control section 23 creates ClipInfo about the above-mentioned AV stream file.

[0526]At Step S203, the control section 23 creates STC\_Info about the above-mentioned AV stream file.

[0527]At Step S204, the control section 23 creates Program\_Info about the above-mentioned AV stream file.

[0528]At Step S205, the control section 23 creates CPI (EP-map or TU-map) about the above-mentioned AV stream file.

[0529]At Step S206, the control section 23 creates ClipMark about the above-mentioned AV stream file.

[0530]At Step S207, the control section 23 records the Clip Information file in which the above-mentioned ClipInfo, STC\_Info, ProgramInfo, CPI, and ClipMark were stored.

[0531]Although each processing was explained to the time series here, Step S206 operates simultaneously actually from Step S201.

[0532]Next, the example of creation of STC\_Info of operation is explained using the flow chart of drawing 14. This processing is performed by the multiplexed stream analyzing parts 18 of drawing 1.

[0533]In Step S221, it is investigated whether PCR packet reception of the stream analyzing parts 18 was carried out. In Step S221, in No, it returns to Step S221, and, in Yes, progresses to Step S222.

[0534]In Step S222, it is investigated whether the discontinuity of STC was detected. In NO, it returns to Step S221. In YES, it progresses to Step S223. In the case of the PCR packet received by the beginning, it certainly progresses to Step S223 after a recording start.

[0535]In Step S223, the number (address) of the transport packet which transmits

PCR of the beginning of new STC is acquired.

[0536]STC\_Info is created in Step S224.

[0537]In Step S225, it is investigated whether the last transport packet carried out the end of an input. In No, it returns to Step S221, and, in Yes, processing is ended.

[0538]The example of creation of Program\_Info of operation is explained using the flow chart of Drawing 142. This processing is performed by the multiplexed stream analyzing parts 18 of drawing 1.

[0539]In Step S241, the stream analyzing parts 18 investigate whether the transport packet containing PSI/SI was received. Here, specifically, the transport packet of PSI/SI is a packet of PAT, PMT, and SIT. SIT is a transport packet the servicing information of the partial transport stream specified by the DVB standard is described to be. In Step 241, in No, it returns to Step S241, and, in Yes, progresses to Step S242.

[0540]In Step S242, it is investigated whether the contents of PSI/SI changed. That is, it is investigated whether each contents of PAT, PMT, and SIT changed compared with each contents received before. When the contents are not changing, it returns to Step S241. When the contents change, it progresses to Step S243. In PSI/SI received by the beginning, it certainly progresses to Step S243 after a recording start.

[0541]In Step S243, the number (address) and the contents of the transport packet which transmits new PSI/SI are acquired.

[0542]The information on Program-sequence is created in Step S244.

[0543]In Step S245, it is investigated whether the last transport packet carried out the end of an input. In No, it returns to Step S241, and, in Yes, processing is ended.

[0544]Next, the example of creation of EP\_map of operation is explained using the flow chart of Drawing 143. This processing is performed by the multiplexed stream analyzing parts 18 of drawing 1.

[0545]The stream analyzing parts 18 set PID of the video of AV program to record at Step S261. When two or more videos are contained in the transport stream, each video PID is set.

[0546]The stream analyzing parts 18 receive the transport packet of video at Step S262.

[0547]Stream analyzing parts investigate whether the pay load (data division following a packet header) of a transport packet has begun from the 1st byte of a PES packet at Step S263 (a PES packet). It is the packet specified by MPEG 2, and an elementary stream is packet-ized. This is understood by investigating the value of "payload\_unit\_start\_indicator" in a transport packet header, and when this value is 1, the pay load of a transport packet starts it from the 1st byte of a PES packet. At Step

S263, in No, it returns to Step 262, and, in Yes, progresses to Step S264.

[0548]Stream analyzing parts investigate whether the pay load of a PES packet has begun from the 1st byte of sequence\_header\_code (they are numerals of "0x000001B3" at 32 bit length) of MPEG video at Step S264. At Step S264, in No, it returns to Step S262, and, in Yes, progresses to Step S265.

[0549]When you progress to Step S265, let the present transport packet be an entry point.

[0550]At Step S266, stream analyzing parts acquire the packet number of the above-mentioned packet, PTS of I picture started from above-mentioned sequence\_header\_code, and PID of the video by which the entry point belongs, and input them into the control section 23. The control section 23 creates EP\_map.

[0551]At Step S267, the present packet judges whether it is a transport packet inputted at the end. When it is not the last packet, it returns to Step S262. Processing is ended when it is the last packet.

[0552]Drawing 144 shows the flow chart explaining the preparation method of ClipMark of the syntax shown in drawing 81, when encoding and recording an analog AV signal.

[0553]At Step S281, the analyzing parts 14 analyze the input AV signal from the terminals 11 and 12, and detect the focus. Specify the characteristic scene resulting from the contents of the AV stream, for example, the focus is a head broth point, a scene change point, etc. of a program.

[0554]At Step S282, the control section 23 acquires PTS of the picture of the focus.

[0555]At Step S283, the control section 23 stores the information on the focus in ClipMark. Specifically, the information explained by the syntax and semantics of ClipMark of this example is stored.

[0556]Drawing 145 shows the flow chart explaining the preparation method of ClipMark of the syntax shown in drawing 81, when recording the transport stream inputted from the digital interface.

[0557]elementalist ream PID of the program which the demultiplexer 26 and the control section 23 record at Step S301 -- acquiring . When there are two or more elementalist reams of an analytical object, all the elementalist ream PID is acquired.

[0558]At Step S302, the demultiplexer 26 separates elementalist ream from the program of the transport stream inputted from the terminal 13, and inputs it into AV decoder 27. AV decoder 27 decodes an input stream and outputs an AV signal.

[0559]At Step S303, the analyzing parts 14 analyze the above-mentioned AV signal, and detect the focus.

[0560]At Step S304, the control section 23 acquires PTS of the picture of the focus, and STC=sequence-id of STC to which it belongs.

[0561]At Step S305, the control section 23 stores the information on the focus in ClipMark. Specifically, the information explained by the syntax and semantics of ClipMark of this example is stored.

[0562]Next, how to perform special reproduction using EP\_map is explained. EP\_map is useful in order to carry out random access reproduction.

[0563]In the transport stream of digital broadcasting, since video PID may change, the decoder needs to get to know PID mapping in the transport stream currently recorded. Therefore, EP\_map has a value of video PID which is called EP\_map\_for\_one\_stream\_PID() and which it refers to for every sub table, and ProgramInfo has the information about PID mapping.

[0564]DVR from which, as for Drawing 146, the value of video PID changes in a transport stream The example of MPEG 2 TS is shown. In this case, EP\_map has a sub table for every video PID.

[0565]Drawing 147 shows the player model in the case of carrying out I picture searches (a trick play, a chapter search, etc.). I picture search is performed at the following step.

[0566]1). First, the file system 112 reads the data of Clip Information file (EP\_map, STC\_Info, ProgramInfo) from the disk 111 (it corresponds to the recording medium 100 of drawing 1). The data is sent to the host controller 115.

2) A user interface sets PTS of the program number to reproduce and search time of onset. And the value is sent to the host controller 115.

3) The host controller 115 sets to the demultiplexer 113 video PID of the source packet which RSPN\_EP\_start corresponding to search time of onset points out.

4) The host controller 115 sets the data address corresponding to the source packet number of said RSPN\_EP\_start to the file system 112.

5) The file system 112 reads a DVR MPEG2 transport stream from the specified data address.

6) When a user sets the following search time, return to the above-mentioned step 2.

[0567]Next, Drawing 148 shows the example of an original AV stream file and the AV stream file after performing edit which eliminates the stream of the partial reproduction range of the stream.

[0568]Before edit, Virtual PlayList assumes that IN\_time and OUT\_time on an original AV stream are pointed out. When edit (minimization edit) which eliminates the stream portion which Virtual PlayList is not using at this time is carried out, it is changed to

the stream after the edit which shows an original AV stream in Drawing 148. The data from the head of an original AV stream to X point and the data from Y point to the last are eliminated. The following explanation explains the example of the method of deciding this X point and Y point.

[0569]Drawing 149 is a figure which illustrates how to eliminate the unnecessary data in front of an IN point without conducting analyzing the contents of the AV stream. PlayList points out the IN point on an original AV stream. EP\_map of the AV stream is illustrated. In order to decode the picture which an IN point points out, I picture started from address ISA2 is required.

[0570]PAT, PMT, and an PCR packet are required after X point. PTS of RSPN\_EP\_start=ISA1 is pts1 and PTS of RSPN\_EP\_start=ISA2 is pts2. If the time lag of the system time base of pts1 and pts2 becomes in more than 100 msec, PAT, PMT, and an PCR packet exist among the addresses ISA1 and ISA2 (at least in SESF, DVB, ATSC, and ISDB, that is right).

[0571]Therefore, X point is decided before address ISA1. And X point must be a boundary of an aligned unit.

[0572]The recorder (recording and reproducing device 1 of drawing 1) can determine X point at the following step using EP\_map without conducting analyzing the contents of the AV stream.

[0573]1) Find RSPN\_EP\_start which has a value of PTS of the past display time in PTS of IN time rather than most the neighborhood and it on system time base.

2) Find RSPN\_EP\_start which has a value of PTS of the display time of the 100msec past at least rather than the value of PTS of RSPN\_EP\_start found at Step 1.

3) X point is decided before RSPN\_EP\_start found at Step 2. And X point must be a boundary of an aligned unit.

[0574]Since this method does not need to read the data of an AV stream and to analyze those contents in order to decide X point, it is easy. However, the AV stream after edit may leave unnecessary data to reproduction of the PlayList. In order to decide X point, supposing it reads the data of an AV stream and analyzes the contents, for reproduction of the PlayList, unnecessary data can be eliminated more efficiently.

[0575]Drawing 150 is a figure which illustrates how to eliminate the unnecessary data behind an OUT point without conducting analyzing the contents of the AV stream. PlayList points out the OUT point on an original AV stream. EP\_map of the AV stream is illustrated.

[0576]The video sequence started from RSPN\_EP\_start=ISA4 is premised on being what is shown below.

I2 B0 B1 P5 ... Here, I, P, and B express I picture, P picture, and B picture, respectively. A number expresses a display order. In this processing, when a recorder does not analyze the contents of the AV stream, the information on the picture which PTS of OUT\_time refers to, including a picture coding type, a temporal reference, etc., does not understand a recorder. When [ this ] the picture B0 or B1 may be referred to (this is not understood when a recorder does not analyze the contents of the AV stream), in order to decode the picture B0 and B1, I2 is required for PTS of OUT\_time. PTS of I2 is larger than PTS of OUT time (pts4 is PTS of I2 in OUT\_time < pts4 and here). Although PTS of I2 is larger than PTS of OUT\_time, I2 is required because of B0 and B1.

[0577]Therefore, Y point is decided behind address ISA5 shown in a figure. ISA5 is a value of RSPN\_EP\_start which is immediately after ISA4 in EP\_map. Y point must be a boundary of an aligned unit again.

[0578]The recorder can determine Y point at the following step using EP\_map without conducting analyzing the contents of the AV stream.

[0579]1) Find RSPN\_EP\_start which has a value of PTS of the display time of the future in PTS of OUT time rather than most the neighborhood and it on system time base.

2) Find RSPN\_EP\_start immediately after RSPN\_EP\_start found at Step 1.

3) Y point is back decided rather than RSPN\_EP\_start found at Step 2. And Y point must be a boundary of an aligned unit.

[0580]Since this method does not need to read the data of an AV stream and to analyze those contents in order to decide Y point, it is easy. However, the AV stream after edit may leave unnecessary data to reproduction of the PlayList. In order to decide Y point, supposing it reads the data of an AV stream and analyzes the contents, for reproduction of the PlayList, unnecessary data can be eliminated more efficiently.

[0581]The contents of the data currently recorded on the recording medium by being based on such syntax, a data structure, and a rule, Reproduction information etc. can be managed appropriately and it has them, and a user can check the contents of the data currently appropriately recorded on the recording medium at the time of reproduction, or it can make it possible to reproduce desired data simple.

[0582]Since according to the database organization of this invention it dissociates independently and a PlayList file and a Clip Information file are recorded, By edit etc., when the contents of a certain PlayList and Clip are changed, it is not necessary to change other files which are unrelated to the file. Therefore, time which can modify a file easily and the change and record take can be made small.

[0583]Read only Info.dvr first and the contents of record of a disk are shown to a user interface, If a user reads only the PlayList file which carried out reproduction instruction, and the Clip Information file relevant to it from a disk, a user's waiting time can be made small.

[0584]If all the PlayList files and Clip Information files are summarized to one file and recorded, the file size will become very large. Therefore, the time which it takes in order to modify the file and to record it becomes very large compared with the case where dissociate independently and each file is recorded. This invention solves this problem.

[0585]As attached information of an AV stream, as mentioned above, ClipInfo of said AV stream, STCInfo which stores the discontinuous dot data of STC of said AV stream, ProgramInfo which stores the discontinuous dot data of the contents of a program of said AV stream, CPI for storing the information which associates the hour entry and address information on said AV stream, ClipMark for storing the address in the type (for example, point of a program pulling out the head) and the AV stream of a picture of the picture of the characteristic picture on said AV stream, By file-izing as Clip Information File and recording on the recording medium 100, it is possible to manage appropriately the encoded information of a stream required for reproduction of a stream required for reproduction of an AV stream.

[0586]Using this Clip Information file information, a user, Can search the scene which is interested out of the AV stream currently recorded on the recording medium 100, for example, the point of a program pulling out the head, etc., and a user's random access and directions of special reproduction are received, The determination of the reading position of the AV stream from the recording medium 100 becomes easy, and the decoding start of a stream can be performed promptly.

[0587]Although a series of processings mentioned above can also be performed by hardware, they can also be performed by software. In this case, for example, the recording and reproducing device 1 is constituted by the personal computer as shown in Drawing 151.

[0588]In Drawing 151, CPU(Central Processing Unit) 201, Various kinds of processings are performed according to the program memorized by ROM(ReadOnly Memory) 202 or the program loaded to RAM(Random Access Memory) 203 from the storage parts store 208. To RAM203, CPU201 performs various kinds of processings again, and also required data etc. are memorized suitably.

[0589]CPU201, ROM202, and RAM203 are mutually connected via the bus 204. The input/output interface 205 is also connected to this bus 204 again.

[0590]The input part 206, CRT which become the input/output interface 205 from a keyboard, a mouse, etc., The communications department 209 which comprises the storage parts store 208, a modem, a terminal adopter, etc. which comprise the outputting part 207 which consists of a display which consists of LCD etc., a loudspeaker, etc., a hard disk, etc. is connected. The communications department 209 performs the communications processing through a network.

[0591]The drive 210 is connected to the input/output interface 205 again if needed, It is suitably equipped with the magnetic disk 221, the optical disc 222, the magneto-optical disc 223, or the semiconductor memory 224, and the computer program read from them is installed in the storage parts store 208 if needed.

[0592]Although a series of processings mentioned above can also be performed by hardware, they can also be performed with software. The computer by which the program which constitutes the software is included in hardware for exclusive use when performing a series of processings with software, Or it is installed in the personal computer etc. which can perform various kinds of functions, for example, are general-purpose, etc. from a recording medium by installing various kinds of programs.

[0593]. As shown in Drawing 151, this recording medium is distributed apart from a computer in order to provide a user with a program. The magnetic disk 221 (a floppy disk is included) with which the program is recorded, the optical disc 222 (CD-ROM (Compact Disk-Read Only Memory).) . DVD (Digital Versatile Disk) is included. It is not only constituted by the package media which consist of the magneto-optical disc 223 (MD (Mini-Disk) is included) or the semiconductor memory 224, but, It comprises a hard disk etc. in which ROM202 with which a user is provided in the state where it was beforehand included in the computer, and the program is remembered to be, and the storage parts store 208 are contained.

[0594]In this specification, even if the processing serially performed according to an order that the step which describes the program provided by a medium was indicated is not of course necessarily processed serially, it also includes a parallel target or the processing performed individually.

[0595]In this specification, a system expresses the whole device constituted by two or more devices.

[0596]

[Effect of the Invention]According to the 1st information processor of this invention and a method, a recording medium, and the program, like the above. As Clip information, the start address information [ \*\*\*\* / the encoded information in an AV stream ] on the section, the information which associates the time information and



address information in an AV stream, and the time information of the characteristic picture in an AV stream are recorded.

[0597]According to the 2nd information processor of this invention and a method, a recording medium, and the program. As Clip information, the start address information [ \*\*\*\* / the encoded information in an AV stream ] on the section, the information which associates the time information and address information in an AV stream, and the time information of the characteristic picture in an AV stream are reproduced.

[0598]Therefore, in the case of which, the determination and decoding processing of a reading position of an AV stream can be performed promptly, and a predetermined mark can be especially searched promptly to it.

---

[Translation done.] \* NOTICES \*

**JPO and INPIT are not responsible for any damages caused by the use of this translation.**

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.\*\*\*\* shows the word which can not be translated.

3.In the drawings, any words are not translated.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1]It is a figure showing the composition of the 1 embodiment of the recording and reproducing device which applied this invention.

[Drawing 2]It is a figure explaining the format of the data recorded on a recording medium by the recording and reproducing device 1.

[Drawing 3]It is a figure explaining Real PlayList and Virtual PlayList.

[Drawing 4]It is a figure explaining creation of Real PlayList.

[Drawing 5]It is a figure explaining deletion of Real PlayList.

[Drawing 6]It is a figure explaining assemble editing.

[Drawing 7]It is a figure explaining the case where a sub path is provided in Virtual

PlayList.

[Drawing 8]It is a figure explaining change of the reproduction sequence of PlayList.

[Drawing 9]It is a figure explaining the mark on PlayList, and the mark on Clip.

[Drawing 10]It is a figure explaining a menu thumbnail.

[Drawing 11]It is a figure explaining the mark added to PlayList.

[Drawing 12]It is a figure explaining the mark added to a clip.

[Drawing 13]It is a figure explaining the relation of PlayList, Clip, and a thumbnail file.

[Drawing 14]It is a figure explaining directory structure.

[Drawing 15]It is a figure showing the syntax of info.dvr.

[Drawing 16]It is a figure showing the syntax of DVR volume.

[Drawing 17]It is a figure showing the syntax of Resumevolume.

[Drawing 18]It is a figure showing the syntax of UIAppInfovolume.

[Drawing 19]It is a figure showing the table of Character set value.

[Drawing 20]It is a figure showing the syntax of TableOfPlayList.

[Drawing 21]It is a figure showing other syntax of TableOfPlayList.

[Drawing 22]It is a figure showing the syntax of MakersPrivateData.

[Drawing 23]xxxxx. It is a figure showing the syntax of rpls and yyyyy.vpls.

[Drawing 24]It is a figure explaining PlayList.

[Drawing 25]It is a figure showing the syntax of PlayList.

[Drawing 26]It is a figure showing the table of PlayList\_type.

[Drawing 27]It is a figure showing the syntax of UIAppinfoPlayList.

[Drawing 28]It is a figure explaining the flag in the syntax of UIAppinfoPlayList shown in drawing 27.

[Drawing 29]It is a figure explaining PlayItem.

[Drawing 30]It is a figure explaining PlayItem.

[Drawing 31]It is a figure explaining PlayItem.

[Drawing 32]It is a figure showing the syntax of PlayItem.

[Drawing 33]It is a figure explaining IN\_time.

[Drawing 34]It is a figure explaining OUT\_time.

[Drawing 35]It is a figure showing the table of Connection\_Condition.

[Drawing 36]It is a figure explaining Connection\_Condition.

[Drawing 37]It is a figure explaining BridgeSequenceInfo.

[Drawing 38]It is a figure showing the syntax of BridgeSequenceInfo.

[Drawing 39]It is a figure explaining SubPlayItem.

[Drawing 40]It is a figure showing the syntax of SubPlayItem.

[Drawing 41]It is a figure showing the table of SubPath\_type.

[Drawing 42]It is a figure showing the syntax of PlayListMark.

[Drawing 43]It is a figure showing the table of Mark\_type.

[Drawing 44]It is a figure explaining Mark\_time\_stamp.

[Drawing 45]It is a figure showing the syntax of zzzzz.clip.

[Drawing 46]It is a figure showing the syntax of ClipInfo.

[Drawing 47]It is a figure showing the table of Clip\_stream\_type.

[Drawing 48]It is a figure explaining offset\_SPN.

[Drawing 49]It is a figure explaining offset\_SPN.

[Drawing 50]It is a figure explaining the STC section.

[Drawing 51]It is a figure explaining STC\_Info.

[Drawing 52]It is a figure showing the syntax of STC\_Info.

[Drawing 53]It is a figure explaining ProgramInfo.

[Drawing 54]It is a figure showing the syntax of ProgramInfo.

[Drawing 55]It is a figure showing the syntax of VideoCondngInfo.

[Drawing 56]It is a figure showing the table of Video\_format.

[Drawing 57]It is a figure showing the table of frame\_rate.

[Drawing 58]It is a figure showing the table of display\_aspect\_ratio.

[Drawing 59]It is a figure showing the syntax of AudioCondngInfo.

[Drawing 60]It is a figure showing the table of audio\_coding.

[Drawing 61]It is a figure showing the table of audio\_component\_type.

[Drawing 62]It is a figure showing the table of sampling\_frequency.

[Drawing 63]It is a figure explaining CPI.

[Drawing 64]It is a figure explaining CPI.

[Drawing 65]It is a figure showing the syntax of CPI.

[Drawing 66]It is a figure showing the table of CPI\_type.

[Drawing 67]It is a figure explaining video EP\_map.

[Drawing 68]It is a figure explaining EP\_map.

[Drawing 69]It is a figure explaining EP\_map.

[Drawing 70]It is a figure showing the syntax of EP\_map.

[Drawing 71]It is a figure showing the table of EP\_type values.

[Drawing 72]It is a figure showing the syntax of EP\_map\_for\_one\_stream\_PID.

[Drawing 73]It is a figure explaining TU\_map.

[Drawing 74]It is a figure showing the syntax of TU\_map.

[Drawing 75]It is a figure showing the syntax of ClipMark.

[Drawing 76]It is a figure showing the table of mark\_type.

[Drawing 77]It is a figure showing the table of mark\_type\_stamp.

[Drawing 78]It is a figure showing other examples of the syntax of ClipMark.

[Drawing 79]It is a figure showing other examples of the table of Mark\_type.

[Drawing 80]It is a figure showing the example of mark\_entry() and representative\_picture\_entry().

[Drawing 81]It is a figure showing the syntax of mark\_entry() and representative\_picture\_entry().

[Drawing 82]It is a figure showing other examples of the syntax of mark\_entry() and representative\_picture\_entry().

[Drawing 83]It is a figure explaining the relation between RSPN\_ref\_EP\_start and offset\_num\_pictures.

[Drawing 84]It is a figure showing other examples of the syntax of mark\_entry() and representative\_picture\_entry().

[Drawing 85]It is a figure explaining the relation between ClipMark and EP\_map.

[Drawing 86]It is a figure showing the syntax of menu.thmb and mark.thmb.

[Drawing 87]It is a figure showing the syntax of Thumbnail.

[Drawing 88]It is a figure showing the table of thumbnail\_picture\_format.

[Drawing 89]It is a figure explaining tn\_block.

[Drawing 90]It is a figure explaining the structure of the transport stream of DVR MPEG 2.

[Drawing 91]It is a figure showing the recorder model of the transport stream of DVR MPEG 2.

[Drawing 92]It is a figure showing the player model of the transport stream of DVR MPEG 2.

[Drawing 93]It is a figure showing the syntax of source packet.

[Drawing 94]It is a figure showing the syntax of TP\_extra\_header.

[Drawing 95]It is a figure showing the table of copy permission indicator.

[Drawing 96]It is a figure explaining seamless connection.

[Drawing 97]It is a figure explaining seamless connection.

[Drawing 98]It is a figure explaining seamless connection.

[Drawing 99]It is a figure explaining seamless connection.

[Drawing 100]It is a figure explaining seamless connection.

[Drawing 101]It is a figure explaining the overlap of an audio.

[Drawing 102]It is a figure explaining the seamless connection using BridgeSequence.

[Drawing 103]It is a figure explaining the seamless connection which does not use BridgeSequence.

[Drawing 104]It is a figure showing a DVR STD model.

[Drawing 105]It is a figure showing the timing chart of decoding and a display.

[Drawing 106]It is a flow chart explaining search reproduction of the scene shown with the marking point in the case of the syntax of drawing 81.

[Drawing 107]It is a figure explaining operation of reproduction in case mark\_entry()/representative\_picture\_entry() of ClipMark of drawing 75 or drawing 78 is the syntax of drawing 81.

[Drawing 108]It is a figure showing the example of EP\_map.

[Drawing 109]It is a figure showing the example of ClipMark.

[Drawing 110]It is a flow chart explaining CM skip regeneration in case mark\_entry()/representative\_picture\_entry() of ClipMark of drawing 75 or drawing 78 is the syntax of drawing 81.

[Drawing 111]It is a flow chart explaining CM skip regeneration in case mark\_entry()/representative\_picture\_entry() of ClipMark of drawing 75 or drawing 78 is the syntax of drawing 81.

[Drawing 112]It is a flow chart explaining search reproduction of the scene shown with a marking point in case mark\_entry()/representative\_picture\_entry() of ClipMark of drawing 75 or drawing 78 is the syntax of drawing 82.

[Drawing 113]It is a figure explaining reproduction in case mark\_entry()/representative\_picture\_entry() of ClipMark of drawing 75 or drawing 78 is the syntax of drawing 82.

[Drawing 114]It is a figure showing the example of EP\_map.

[Drawing 115]It is a figure showing the example of ClipMark.

[Drawing 116]It is a flow chart explaining CM skip reproduction in case mark\_entry()/representative\_picture\_entry() of ClipMark of drawing 75 or drawing 78 is the syntax of drawing 82.

[Drawing 117]It is a flow chart explaining CM skip reproduction in case mark\_entry()/representative\_picture\_entry() of ClipMark of drawing 75 or drawing 78 is the syntax of drawing 82.

[Drawing 118]It is a flow chart explaining search reproduction of the scene shown with a marking point in case mark\_entry()/representative\_picture\_entry() of ClipMark of drawing 75 or drawing 78 is the syntax of drawing 84.

[Drawing 119]It is a figure explaining reproduction in case mark\_entry()/representative\_picture\_entry() of ClipMark of drawing 75 or drawing 78 is the syntax of drawing 84.

[Drawing 120]It is a figure showing the example of EP\_map.

[Drawing 121]It is a figure showing the example of ClipMark.

[Drawing 122] It is a flow chart explaining CM skip reproduction in case mark\_entry()/representative\_picture\_entry() of ClipMark of drawing 75 or drawing 78 is the syntax of drawing 84.

[Drawing 123] It is a flow chart explaining CM skip reproduction in case mark\_entry()/representative\_picture\_entry() of ClipMark of drawing 75 or drawing 78 is the syntax of drawing 84.

[Drawing 124] It is a figure showing an application format.

[Drawing 125] It is a figure explaining the mark on PlayList, and the mark on Clip.

[Drawing 126] It is a figure showing other examples of the syntax of ClipMark.

[Drawing 127] It is a figure showing the example of further others of the syntax of ClipMark.

[Drawing 128] It is a figure showing another example of the syntax of ClipInfo().

[Drawing 129] It is a figure showing another example of the syntax of ProgramInfo().

[Drawing 130] It is a figure showing the syntax of StreamCodingInfo().

[Drawing 131] It is a figure explaining stream\_coding\_type.

[Drawing 132] It is a figure explaining the relation between EP-fine and EP-coarse.

[Drawing 133] It is a figure explaining the format of PTS\_EP\_fine and PTS\_EP\_coarse.

[Drawing 134] It is a figure explaining the format of RSPN\_EP\_fine and RSPN\_EP\_coarse.

[Drawing 135] It is a figure explaining the entry of EP-coarse, and the entry of EP-fine.

[Drawing 136] It is a figure showing another example of the syntax of EP\_map.

[Drawing 137] It is a figure explaining EP\_stream\_type values.

[Drawing 138] It is a figure showing the syntax of EP\_map\_for\_one\_stream\_PID of EP\_map of Drawing 136.

[Drawing 139] It is a figure explaining the meaning of the value of EP\_video\_type.

[Drawing 140] It is a flow chart explaining creation processing of a Clip AV stream file and a Clip Information file.

[Drawing 141] It is a flow chart explaining the example of creation of STC\_Info of operation.

[Drawing 142] It is a flow chart explaining the example of creation of ProgramInfo of operation.

[Drawing 143] It is a flow chart explaining the example of creation of EP\_map of operation.

[Drawing 144]. It can set, when encoding and recording an analog AV signal. It is a flow chart explaining the preparation method of ClipMark in case mark\_entry()/representative\_picture\_entry() of ClipMark of drawing 75 or drawing 78 is

the syntax shown in drawing 81.

[Drawing 145]. It can set, when recording the transport stream inputted from the digital interface. It is a flow chart explaining the preparation method of ClipMark in case mark\_entry()/representative\_picture\_entry() of ClipMark of drawing 75 or drawing 78 is the syntax shown in drawing 81.

[Drawing 146]It is a figure explaining the special reproduction using EP\_map.

[Drawing 147]It is a figure explaining the player model for I picture search which uses EP\_map.

[Drawing 148]It is a figure showing the example of the operation of minimization.

[Drawing 149]It is a figure showing the example which eliminates the unnecessary stream data in front of IN\_time at the time of minimization.

[Drawing 150]It is a figure explaining the example which eliminates the unnecessary stream data behind OUT\_time at the time of minimization.

[Drawing 151]It is a figure explaining a medium.

[Description of Notations]

1 A recording and reproducing device, and 11 thru/or 13 A terminal, 14 Analyzing parts and 15 AV encoder, 16 A multiplexer and 17 A switch, 18 Multiplexed stream analyzing parts and 19 [ A user interface and 26 / A demultiplexer and 27 AV decoders, ] Saw spa KETTAIZA, 20 ECC-code-ized part, and 21 A modulation part, 22 writing parts, and 23 A control section and 24 28 A read section, 29 demodulation sections, 30 ECC decoding part, and 31 Saw spa KETTAIZA, 32, and 33 Terminal

---

[Translation done.]

(19)日本国特許庁（J P）(12)公開特許公報（A）(11)特許出願公開番号  
特開2002－158971  
（P2002－158971A）  
(43)公開日 平成14年 5 月31日 (2002. 5. 31)

(51)Int.Cl.<sup>7</sup>  
H 0 4 N 5/92  
G 1 1 B 20/12  
  
27/00  
H 0 4 N 5/91

識別記号  
  
  
1 0 3

F I  
G 1 1 B 20/12  
  
27/00  
H 0 4 N 5/92  
5/91

テーマコード\* (参考)  
5 C 0 5 3  
1 0 3 5 D 0 4 4  
D 5 D 1 1 0  
H  
N

審査請求 未請求 請求項の数19 O L （全 75 頁）

(21)出願番号	特願2001－89959(P2001－89959)	(71)出願人	000002185 ソニー株式会社 東京都品川区北品川 6 丁目 7 番35号
(22)出願日	平成13年 3 月27日 (2001. 3. 27)	(72)発明者	加藤 元樹 東京都品川区北品川 6 丁目 7 番35号 ソニ ー株式会社内
(31)優先権主張番号	特願2000－183770(P2000－183770)	(72)発明者	浜田 俊也 東京都品川区北品川 6 丁目 7 番35号 ソニ ー株式会社内
(32)優先日	平成12年 4 月21日 (2000. 4. 21)	(74)代理人	100082131 弁理士 稲本 義雄
(33)優先権主張国	日本（J P）		
(31)優先権主張番号	特願2000－268042(P2000－268042)		
(32)優先日	平成12年 9 月 5 日 (2000. 9. 5)		
(33)優先権主張国	日本（J P）		

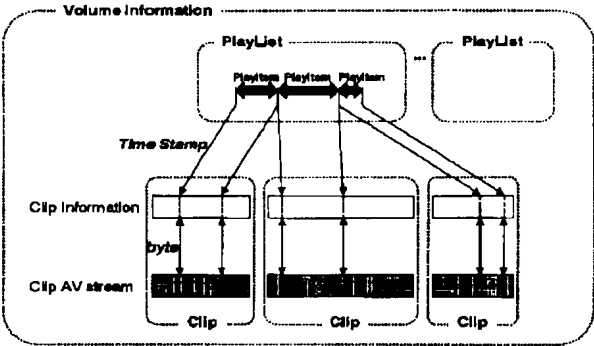
最終頁に続く

(54)【発明の名称】 情報処理装置および方法、記録媒体、プログラム、並びに記録媒体

(57)【要約】

【課題】 所定のマークを迅速に検索できるようにする。

【解決手段】 コンテンツの実体としてのAVストリームは、ClipInformationにより管理され、AVストリームの再生は、PlayListにより管理される。AVストリームの属性情報としての、AVストリーム中の不連続点のアドレス情報RSPN\_arrival\_time\_discontinuity、AVストリーム中の時刻情報とアドレス情報を関連づける情報EP\_map、TU\_map、並びに、AVストリーム中の特徴的な画像の時刻情報ClipMarkは、ClipInformationに記録される。





## 【特許請求の範囲】

【請求項1】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録される記録媒体に対して前記AVストリームを記録する情報処理装置において、前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、

前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を生成する生成手段と、

前記生成手段により生成された情報を前記Clip情報として前記記録媒体に記録する記録手段とを備えることを特徴とする情報処理装置。

【請求項2】 前記符号化情報が連続な区間の開始アドレス情報は、STCシーケンスまたはプログラムシーケンスの開始アドレスであり、

前記時刻情報とアドレス情報を関連づける情報は、EP\_mapまたはTU\_mapであり、

前記特徴的な画像の時刻情報は、ClipMarkであることを特徴とする請求項1に記載の情報処理装置。

【請求項3】 前記記録手段は、前記AVストリームの記録レートの平均値に関する情報を前記記録媒体にさらに記録することを特徴とする請求項1に記載の情報処理装置。

【請求項4】 前記平均値に関する情報は、TS\_average\_rateであることを特徴とする請求項3に記載の情報処理装置。

【請求項5】 前記AVストリームはトランスポートストリームであることを特徴とする請求項1に記載の情報処理装置。

【請求項6】 前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、前記トランスポートストリームのシステムタイムクロックが連続な区間であるSTCシーケンスの開始アドレスを含むことを特徴とする請求項1に記載の情報処理装置。

【請求項7】 前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、前記トランスポートストリームのプログラム内容が一定な区間であるプログラムシーケンスの開始アドレスを含むことを特徴とする請求項1に記載の情報処理装置。

【請求項8】 前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、前記トランスポートストリームのトランスポートパケットの到着時間に基づくアライバルタイムが連続な区間の開始アドレスを含むことを特徴とする請求項1に記載の情報処理装置。

【請求項9】 前記AVストリーム中の時刻情報とアドレス情報を関連づける情報は、前記トランスポートストリームのIピクチャのアドレスとそのプレゼンテーションタイムスタンプを含むことを特徴とする請求項1に記載の情報処理装置。

【請求項10】 前記AVストリーム中の時刻情報とアドレス情報を関連づける情報の少なくとも一部を圧縮する圧縮手段をさらに備え、

前記記録手段は、前記圧縮手段により圧縮された前記情報を記録することを特徴とする請求項9に記載の情報処理装置。

【請求項11】 前記AVストリーム中の時刻情報とアドレス情報を関連づける情報は、トランスポートパケットの到着時刻に基づいたアライバルタイムと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスを含むことを特徴とする請求項1に記載の情報処理装置。

【請求項12】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録される記録媒体に対して前記AVストリームを記録する情報処理装置の情報処理方法において、

前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、

前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、

前記生成ステップにより生成された情報を前記Clip情報として前記記録媒体に記録する記録ステップとを含むことを特徴とする情報処理方法。

【請求項13】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録される記録媒体に対して前記AVストリームを記録する情報処理装置のプログラムにおいて、

前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、

前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、前記生成ステップにより生成された情報を前記Clip情報として前記記録媒体に記録する記録ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項14】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録される記録媒体に対して前記AVストリームを記録する情報処理装置を制御するコンピュータに、

前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、

前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、

前記生成ステップにより生成された情報を前記Clip情報

として前記記録媒体に記録する記録ステップとを実行させるプログラム。

【請求項 15】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録されている記録媒体から前記AVストリームを再生する情報処理装置において、前記Clip情報として、前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を再生する再生手段と、前記再生手段により再生された前記Clip情報に基づいて、前記AVストリームの出力を制御する制御手段とを備えることを特徴とする情報処理装置。

【請求項 16】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録されている記録媒体から前記AVストリームを再生する情報処理装置の情報処理方法において、前記Clip情報として、前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、前記再生ステップの処理により再生された前記Clip情報に基づいて、前記AVストリームの出力を制御する制御ステップとを含むことを特徴とする情報処理方法。

【請求項 17】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録されている記録媒体から前記AVストリームを再生する情報処理装置のプログラムにおいて、前記Clip情報として、前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、前記再生ステップの処理により再生された前記Clip情報に基づいて、前記AVストリームの出力を制御する制御ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項 18】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録されている記録媒体から前記AVストリームを再生する情報処理装置を制御するコンピュータに、前記Clip情報として、前記AVストリーム中の符号化情報が連続な区間の開始ア

ドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、前記再生ステップの処理により再生された前記Clip情報に基づいて、前記AVストリームの出力を制御する制御ステップとを実行させるプログラム。

【請求項 19】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録されている記録媒体において、前記Clip情報として、前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報が記録されていることを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は情報処理装置および方法、記録媒体、並びにプログラムに関し、特に、AVストリーム内のIピクチャのアドレス情報、符号化パラメータ、変化点情報、マークなどの情報をファイルとして記録する情報処理装置および方法、記録媒体、並びにプログラムに関する。

【0002】

【従来の技術】近年、記録再生装置から取り外し可能なディスク型の記録媒体として、各種の光ディスクが提案されつつある。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のAV(Audio Visual)信号を記録するメディアとしての期待が高い。この記録可能な光ディスクに記録するデジタルのAV信号のソース（供給源）としては、CSデジタル衛星放送やBSデジタル放送があり、また、将来はデジタル方式の地上波テレビジョン放送等も提案されている。

【0003】ここで、これらのソースから供給されるデジタルビデオ信号は、通常MPEG(Moving Picture Experts Group)2方式で画像圧縮されているのが一般的である。また、記録装置には、その装置固有の記録レートが定められている。従来の民生用映像蓄積メディアで、デジタル放送のデジタルビデオ信号を記録する場合、アナログ記録方式であれば、デジタルビデオ信号をデコード後、帯域制限をして記録する。あるいは、MPEG1 Video、MPEG2 Video、DV方式をはじめとするデジタル記録方式であれば、1度デコードされた後に、その装置固有の記録レート・符号化方式で再エンコードされて記録される。

【0004】しかしながら、このような記録方法は、供給されたビットストリームを1度デコードし、その後で帯域制限や再エンコードを行って記録するため、画質の

劣化を伴う。画像圧縮されたデジタル信号の記録をする場合、入力されたデジタル信号の伝送レートが記録再生装置の記録レートを超えない場合には、供給されたビットストリームをデコードや再エンコードすることなく、そのまま記録する方法が最も画質の劣化が少ない。ただし、画像圧縮されたデジタル信号の伝送レートが記録媒体としてのディスクの記録レートを超える場合には、記録再生装置でデコード後、伝送レートがディスクの記録レートの上限以下になるように、再エンコードをして記録する必要はある。

【0005】また、入力デジタル信号のビットレートが時間により増減する可変レート方式によって伝送されている場合には、回転ヘッドが固定回転数であるために記録レートが固定レートになるテープ記録方式に比べ、1度バッファにデータを蓄積し、バースト的に記録ができるディスク記録装置の方が記録媒体の容量をより無駄なく利用できる。

【0006】以上のように、デジタル放送が主流となる将来においては、データストリーマのように放送信号をデジタル信号のまま、デコードや再エンコードすることなく記録し、記録媒体としてディスクを使用した記録再生装置が求められると予測される。

【0007】

【発明が解決しようとする課題】上述したような装置により、複数のデータ（例えば、映像データや音声データなどから構成される番組のデータ）が記録されている記録媒体を再生する際、ユーザのランダムアクセスや特殊再生の指示に対して、記録媒体からのAVストリームの読み出し位置の決定やストリームの復号といった処理を速やかに行わなくてはならないが、記録媒体に記録されるデータ量が増加するに従い、そのような処理を速やかに行えないといった課題があった。

【0008】本発明はこのような状況に鑑みてなされたものであり、AVストリーム内の1ピクチャのアドレス情報、符号化パラメータ、変化点情報、マークなどの情報をファイルとして記録することにより、AVストリームの読み出し位置の決定や復号処理を速やかに行えるようにし、特に、所定のマークを迅速に検索できるようにすることを目的とする。

【0009】

【課題を解決するための手段】本発明の第1の情報処理装置は、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を生成する生成手段と、生成手段により生成された情報をClip情報として記録媒体に記録する記録手段とを備えることを特徴とする。

【0010】前記符号化情報が連続な区間の開始アドレス情報は、STCシーケンスまたはプログラムシーケンスの開始アドレスであり、時刻情報とアドレス情報を関連

づける情報は、EP\_mapまたはTU\_mapであり、特徴的な画像の時刻情報は、ClipMarkとすることができる。

【0011】前記記録手段は、AVストリームの記録レートの平均値に関する情報を記録媒体にさらに記録することができる。

【0012】前記平均値に関する情報は、TS\_average\_rateとすることができる。

【0013】前記AVストリームはトランスポートストリームとすることができる。

10 【0014】前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、トランスポートストリームのシステムタイムクロックが連続な区間であるSTCシーケンスの開始アドレスを含むようにすることができる。

【0015】前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、トランスポートストリームのプログラム内容が一定な区間であるプログラムシーケンスの開始アドレスを含むようにすることができる。

20 【0016】前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、トランスポートストリームのトランスポートパケットの到着時間に基づくアライバルタイムが連続な区間の開始アドレスを含むようにすることができる。

【0017】前記AVストリーム中の時刻情報とアドレス情報を関連づける情報は、トランスポートストリームの1ピクチャのアドレスとそのプレゼンテーションタイムスタンプを含むようにすることができる。

【0018】前記AVストリーム中の時刻情報とアドレス情報を関連づける情報の少なくとも一部を圧縮する圧縮手段をさらに備え、記録手段は、圧縮手段により圧縮された情報を記録することができる。

30 【0019】前記AVストリーム中の時刻情報とアドレス情報を関連づける情報は、トランスポートパケットの到着時刻に基づいたアライバルタイムと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスを含むようにすることができる。

【0020】本発明の第1の情報処理方法は、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、生成ステップにより生成された情報をClip情報として記録媒体に記録する記録ステップとを含むことを特徴とする。

【0021】本発明の第1の記録媒体のプログラムは、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、生成ステップにより生成された情報をClip情報として記録媒体に記録する記録ステップとを含むことを特徴とする。

50 【0022】本発明の第1のプログラムは、AVストリー

ム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、生成ステップにより生成された情報をClip情報として記録媒体に記録する記録ステップとを実行させる。

【0023】本発明の第2の情報処理装置は、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を再生する再生手段と、再生手段により再生されたClip情報に基づいて、AVストリームの出力を制御する制御手段とを備えることを特徴とする。

【0024】本発明の第2の情報処理方法は、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、再生ステップの処理により再生されたClip情報に基づいて、AVストリームの出力を制御する制御ステップとを含むことを特徴とする。

【0025】本発明の第2の記録媒体のプログラムは、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、再生ステップの処理により再生されたClip情報に基づいて、AVストリームの出力を制御する制御ステップとを含むことを特徴とする。

【0026】本発明の第2のプログラムは、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、再生ステップの処理により再生されたClip情報に基づいて、AVストリームの出力を制御する制御ステップとを実行させるプログラム。

【0027】本発明の記録媒体は、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報が記録されていることを特徴とする。

【0028】本発明の第1の情報処理装置および方法、記録媒体、並びにプログラムにおいては、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報が記録される。

【0029】本発明の第2の情報処理装置および方法、記録媒体、並びにプログラムにおいては、Clip情報とし

て、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報が再生される。

【0030】

【発明の実施の形態】以下に、本発明の実施の形態について、図面を参照して説明する。図1は、本発明を適用した記録再生装置1の内部構成例を示す図である。まず、外部から入力された信号を記録媒体に記録する動作を行う記録部2の構成について説明する。記録再生装置1は、アナログデータ、または、デジタルデータを入力し、記録することができる構成とされている。

【0031】端子11には、アナログのビデオ信号が、端子12には、アナログのオーディオ信号が、それぞれ入力される。端子11に入力されたビデオ信号は、解析部14とAVエンコーダ15に、それぞれ出力される。端子12に入力されたオーディオ信号は、解析部14とAVエンコーダ15に出力される。解析部14は、入力されたビデオ信号とオーディオ信号からシーンチェンジなどの特徴点を抽出する。

【0032】AVエンコーダ15は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム(V)、符号化オーディオストリーム(A)、およびAV同期等のシステム情報(S)をマルチプレクサ16に出力する。

【0033】符号化ビデオストリームは、例えば、MPEG(Moving Picture Expert Group)2方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、MPEG1方式により符号化されたオーディオストリームや、ドルビーAC3方式(商標)により符号化されたオーディオストリーム等である。マルチプレクサ16は、入力されたビデオおよびオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ17を介して多重化ストリーム解析部18とソースパケットタイザ19に出力する。

【0034】多重化ストリームは、例えば、MPEG2トランスポートストリームやMPEG2プログラムストリームである。ソースパケットタイザ19は、入力された多重化ストリームを、そのストリームを記録させる記録媒体100のアプリケーションフォーマットに従って、ソースパケットから構成されるAVストリームに符号化する。AVストリームは、ECC(誤り訂正)符号化部20と変調部21でECC符号の付加と変調処理が施され、書き込み部22に出力される。書き込み部22は、制御部23から出力される制御信号に基づいて、記録媒体100にAVストリームファイルを書き込む(記録する)。

【0035】デジタルインタフェースまたはデジタルテレビジョンチューナから入力されるデジタルテレビジョン放送等のトランスポートストリームは、端子13に入力される。端子13に入力されたトランスポートストリ

ームの記録方式には、2通りあり、それらは、トランスペアレントに記録する方式と、記録ビットレートを下げるなどの目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインターフェースとしての端子24から制御部23へ入力される。

【0036】入力トランスポートストリームをトランスペアレントに記録する場合、端子13に入力されたトランスポートストリームは、スイッチ17を介して多重化ストリーム解析部18と、ソースパケットタイザ19に出力される。これ以降の記録媒体100へAVストリームが記録されるまでの処理は、上述のアナログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【0037】入力トランスポートストリームを再エンコードした後に記録する場合、端子13に入力されたトランスポートストリームは、デマルチプレクサ26に輸入される。デマルチプレクサ26は、入力されたトランスポートストリームに対してデマルチプレクス処理を施し、ビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)を抽出する。

【0038】デマルチプレクサ26により抽出されたストリーム(情報)のうち、ビデオストリームはAVデコーダ27に、オーディオストリームとシステム情報はマルチプレクサ16に、それぞれ出力される。AVデコーダ27は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ15に出力する。AVエンコーダ15は、入力ビデオ信号を符号化し、符号化ビデオストリーム(V)をマルチプレクサ16に出力する。

【0039】一方、デマルチプレクサ26から出力され、マルチプレクサ16に輸入されたオーディオストリームとシステム情報、および、AVエンコーダ15から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部18とソースパケットタイザ19にスイッチ17を介して出力される。これ以後の記録媒体100へAVストリームが記録されるまでの処理は、上述のアナログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【0040】本実施の形態の記録再生装置1は、AVストリームのファイルを記録媒体100に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部23により作成される。制御部23への入力情報は、解析部14からの動画像の特徴情報、多重化ストリーム解析部18からのAVストリームの特徴情報、および端子24から入力されるユーザからの指示情報である。

【0041】解析部14から供給される動画像の特徴情報は、AVエンコーダ15がビデオ信号を符号化する場合

において、解析部14により生成されるものである。解析部14は、入力ビデオ信号とオーディオ信号の内容を解析し、入力動画像信号の中の特徴的な画像(クリップマーク)に関係する情報を生成する。これは、例えば、入力ビデオ信号の中のプログラムの開始点、シーンチェンジ点やCMコマercialのスタート点・エンド点、タイトルやテロップなどの特徴的なクリップマーク点の画像の指示情報であり、また、それにはその画像のサムネイルも含まれる。さらにオーディオ信号のステレオとモノラルの切り換え点や、無音区間などの情報も含まれる。

【0042】これらの画像の指示情報は、制御部23を介して、マルチプレクサ16へ入力される。マルチプレクサ16は、制御部23からクリップマークとして指定される符号化ピクチャを多重化する時に、その符号化ピクチャをAVストリーム上で特定するための情報を制御部23に返す。具体的には、この情報は、ピクチャのPTS(プレゼンテーションタイムスタンプ)またはその符号化ピクチャのAVストリーム上でのアドレス情報である。制御部23は、特徴的な画像の種類とその符号化ピクチャをAVストリーム上で特定するための情報を関連付けて記憶する。

【0043】多重化ストリーム解析部18からのAVストリームの特徴情報は、記録されるAVストリームの符号化情報に関係する情報であり、解析部18により生成される。例えば、AVストリーム内のIピクチャのタイムスタンプとアドレス情報、システムタイムクロックの不連続点情報、AVストリームの符号化パラメータ、AVストリームの中の符号化パラメータの変化点情報などが含まれる。また、端子13から入力されるトランスポートストリームをトランスペアレントに記録する場合、多重化ストリーム解析部18は、入力トランスポートストリームの中から前出のクリップマークの画像を検出し、その種類とクリップマークで指定するピクチャを特定するための情報を生成する。

【0044】端子24からのユーザの指示情報は、AVストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリジューム点の情報などである。

【0045】制御部23は、上記の入力情報に基づいて、AVストリームのデータベース(Clip)、AVストリームの再生区間(PlayItem)をグループ化したもの(Playlist)のデータベース、記録媒体100の記録内容の管理情報(info.dvr)、およびサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、AVストリームと同様に、ECC符号化部20、変調部21で処理されて、書き込み部22へ入力される。書き込み部22は、制御部23から出力される制御信号に基づいて、記録媒体100へデータベースファイルを記録する。

10

20

30

40

50

【0046】上述したアプリケーションデータベース情報についての詳細は後述する。

【0047】このようにして記録媒体100に記録されたAVストリームファイル（画像データと音声データのファイル）と、アプリケーションデータベース情報が再生部3により再生される場合、まず、制御部23は、読み出し部28に対して、記録媒体100からアプリケーションデータベース情報を読み出すように指示する。そして、読み出し部28は、記録媒体100からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部29とECC復号部30の復調と誤り訂正処理を経て、制御部23へ入力される。

【0048】制御部23は、アプリケーションデータベース情報に基づいて、記録媒体100に記録されているPlayListの一覧を端子24のユーザインターフェースへ出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部23へ入力される。制御部23は、そのPlayListの再生に必要なAVストリームファイルの読み出しを、読み出し部28に指示する。読み出し部28は、その指示に従い、記録媒体100から対応するAVストリームを読み出し復調部29に出力する。復調部29に入力されたAVストリームは、所定の処理が施されることにより復調され、さらにECC復号部30の処理を経て、ソースデパケッタ31出力される。

【0049】ソースデパケッタ31は、記録媒体100から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチプレクサ26が処理可能なストリームに変換する。デマルチプレクサ26は、制御部23により指定されたAVストリームの再生区間(PlayItem)を構成するビデオストリーム(V)、オーディオストリーム(A)、およびAV同期等のシステム情報(S)を、AVデコーダ27に出力する。AVデコーダ27は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子32と端子33から出力する。

【0050】また、ユーザインターフェースとしての端子24から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部23は、AVストリームのデータベース(Clip)の内容に基づいて、記憶媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを、読み出し部28に指示する。例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制御部23は、指定された時刻に最も近いタイムスタンプを持つIピクチャからのデータを読み出すように読み出し部28に指示する。

【0051】また、Clip Informationの中のClipMarkにストアされている番組の頭出し点やシーンチェンジ点の中から、ユーザがあるクリップマークを選択した時（例

例えば、この動作は、ClipMarkにストアされている番組の頭出し点やシーンチェンジ点のサムネイル画像リストをユーザインターフェースに表示して、ユーザが、その中からある画像を選択することにより行われる）、制御部23は、Clip Informationの内容に基づいて、記録媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを読み出し部28へ指示する。すなわち、ユーザが選択した画像がストアされているAVストリーム上でのアドレスに最も近いアドレスにあるIピクチャからのデータを読み出すように読み出し部28へ指示する。読み出し部28は、指定されたアドレスからデータを読み出し、読み出されたデータは、復調部29、ECC復号部30、ソースデパケッタ31の処理を経て、デマルチプレクサ26へ入力され、AVデコーダ27で復号されて、マーク点のピクチャのアドレスで示されるAVデータが再生される。

【0052】また、ユーザによって高速再生(Fast-forward playback)が指示された場合、制御部23は、AVストリームのデータベース(Clip)に基づいて、AVストリームの中のI-ピクチャデータを順次連続して読み出すように読み出し部28に指示する。

【0053】読み出し部28は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

【0054】次に、ユーザが、記録媒体100に記録されているAVストリームの編集をする場合を説明する。ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組Aという歌番組から歌手Aの部分を再生し、その後続けて、番組Bという歌番組の歌手Aの部分を再生したいといった再生経路を作成したい場合、ユーザインターフェースとしての端子24から再生区間の開始点（イン点）と終了点（アウト点）の情報が制御部23に入力される。制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの（PlayList）のデータベースを作成する。

【0055】ユーザが、記録媒体100に記録されているAVストリームの一部を消去したい場合、ユーザインターフェースとしての端子24から消去区間のイン点とアウト点の情報が制御部23に入力される。制御部23は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不要なストリーム部分を消去するように、書き込み部22に指示する。

【0056】ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、かつ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部23は、AVストリームの再生区間(PlayI

10

20

30

40

50

tem)をグループ化したもの(Playlist)のデータベースを作成し、さらに、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

【0057】まず、端子24から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部23へ入力される。制御部23は、読み出し部28にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示する。そして、読み出し部28は、記録媒体100からデータを読み出し、そのデータは、復調部29、ECC復号部30、ソースデパケッタ31を経て、デマルチプレクサ26に出力される。

【0058】制御部23は、デマルチプレクサ26に入力されたデータを解析して、ビデオストリームの再エンコード方法(picture\_coding\_typeの変更、再エンコードする符号化ビット量の割り当て)と、再多重化方式を決定し、その方式をAVエンコーダ15とマルチプレクサ16に供給する。

【0059】次に、デマルチプレクサ26は、入力されたストリームをビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)に分離する。ビデオストリームは、AVデコーダ27に入力されるデータとマルチプレクサ16に入力されるデータがある。前者のデータは、再エンコードするために必要なデータであり、これはAVデコーダ27で復号され、復号されたピクチャはAVエンコーダ15で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ16に入力される。

【0060】マルチプレクサ16は、制御部23から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、ECC符号化部20、変調部21で処理されて、書き込み部22に入力される。書き込み部22は、制御部23から供給される制御信号に基づいて、記録媒体100にAVストリームを記録する。

【0061】以下に、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図2は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、AVストリームの管理のためにPlaylistとClipの2つのレイヤをもつ。Volume Informationは、ディスク内のすべてのClipとPlaylistの管理をする。ここでは、1つのAVストリームとその付属情報のペアを1つのオブジェクトと考え、それをClipと称する。AVストリームファイルはClip AV stream fileと称し、その付属情報は、Clip Information fileと称する。

【0062】1つのClip AV stream fileは、MPEG2トランスポートストリームをアプリケーションフォーマット

によって規定される構造に配置したデータをストアする。一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコンテンツは、時間軸上に展開され、Clipの中のエンタリーポイント(1ピクチャ)は、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられた時、Clip Information fileは、Clip AV stream fileの中でデータの読み出しを開始すべきアドレス情報を見つけるために役立つ。

10 【0063】Playlistについて、図3を参照して説明する。Playlistは、Clipの中からユーザが見たい再生区間を選択し、それを簡単に編集することができるようにするために設けられている。1つのPlaylistは、Clipの中の再生区間の集まりである。所定のClipの中の1つの再生区間は、PlayItemと呼ばれ、それは、時間軸上のイン点(IN)とアウト点(OUT)の対で表される。従って、Playlistは、複数のPlayItemが集まることにより構成される。

20 【0064】Playlistには、2つのタイプがある。1つは、Real Playlistであり、もう1つは、Virtual Playlistである。Real Playlistは、それが参照しているClipのストリーム部分を共有している。すなわち、Real Playlistは、その参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real Playlistが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

30 【0065】Virtual Playlistは、Clipのデータを共有していない。従って、Virtual Playlistが変更または消去されたとしても、Clipの内容には何も変化が生じない。

【0066】次に、Real Playlistの編集について説明する。図4(A)は、Real Playlistのクリエイト(create:作成)に関する図であり、AVストリームが新しいClipとして記録される場合、そのClip全体を参照するReal Playlistが新たに作成される操作である。

40 【0067】図4(B)は、Real Playlistのディバイド(divide:分割)に関する図であり、Real Playlistが所望な点で分けられて、2つのReal Playlistに分割される操作である。この分割という操作は、例えば、1つのPlaylistにより管理される1つのクリップ内に、2つの番組が管理されているような場合に、ユーザが1つ1つの番組として登録(記録)し直したいといったようなときに行われる。この操作により、Clipの内容が変更される(Clip自体が分割される)ことはない。

50 【0068】図4(C)は、Real Playlistのコンバイン(combine:結合)に関する図であり、2つのReal Playlistを結合して、1つの新しいReal Playlistにする操作である。この結合という操作は、例えば、ユーザが2つの番組を1つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更され

る (Clip自体が1つにされる) ことはない。

【0069】図5 (A) は、Real Playlist全体のデリート(delete: 削除)に関する図であり、所定のReal Playlist全体を消去する操作がされた場合、削除されたReal Playlistが参照するClipの、対応するストリーム部分も削除される。

【0070】図5 (B) は、Real Playlistの部分的な削除に関する図であり、Real Playlistの所望な部分が削除された場合、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

【0071】図5 (C) は、Real Playlistのミニマイズ(Minimize: 最小化)に関する図であり、Real Playlistに対応するPlayItemを、Virtual Playlistに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual Playlist にとって不必要なClipの、対応するストリーム部分は削除される。

【0072】上述したような操作により、Real Playlistが変更されて、そのReal Playlistが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual Playlistが存在し、そのVirtual Playlistにおいて、削除されたClipにより問題が生じる可能性がある。

【0073】そのようなことが生じないように、ユーザーに、削除という操作に対して、「そのReal Playlistが参照しているClipのストリーム部分を参照しているVirtual Playlistが存在し、もし、そのReal Playlistが消去されると、そのVirtual Playlistもまた消去されることになるが、それでも良いか?」といったメッセージなどを表示させることにより、確認(警告)を促した後に、ユーザーの指示により削除の処理を実行、または、キャンセルする。または、Virtual Playlistを削除する代わりに、Real Playlistに対してミニマイズの操作が行われるようにする。

【0074】次にVirtual Playlistに対する操作について説明する。Virtual Playlistに対して操作が行われたとしても、Clipの内容が変更されることはない。図6は、アセンブル(Assemble) 編集 (IN-OUT 編集)に関する図であり、ユーザーが見たいと所望した再生区間のPlayItemを作り、Virtual Playlistを作成するといった操作である。PlayItem間のシームレス接続が、アプリケーションフォーマットによりサポートされている(後述)。

【0075】図6 (A) に示したように、2つのReal Playlist 1, 2と、それぞれのReal Playlistに対応するClip 1, 2が存在している場合に、ユーザーがReal Playlist 1内の所定の区間(In1乃至Out1までの区間: PlayItem1)を再生区間として指示し、続けて再生する区間として、Real Playlist 2内の所定の区間(In2乃至Out2までの区間: PlayItem2)を再生区間として指示したとき、図6 (B) に示すように、PlayItem1とPlayItem

2から構成される1つのVirtual Playlistが作成される。

【0076】次に、Virtual Playlistの再編集(Re-editing)について説明する。再編集には、Virtual Playlistの中のイン点やアウト点の変更、Virtual Playlistへの新しいPlayItemの挿入(insert)や追加(append)、Virtual Playlistの中のPlayItemの削除などがある。また、Virtual Playlistそのものを削除することもできる。

【0077】図7は、Virtual Playlistへのオーディオのアフレコ(Audio dubbing (post recording))に関する図であり、Virtual Playlistへのオーディオのアフレコをサブパスとして登録する操作のことである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlaylistのメインパスのAVストリームに、付加的なオーディオストリームが、サブパスとして付加される。

【0078】Real PlaylistとVirtual Playlistで共通の操作として、図8に示すようなPlaylistの再生順序の変更(Moving)がある。この操作は、ディスク(ボリューム)の中でのPlaylistの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of Playlist (図20などを参照して後述する) によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

【0079】次に、マーク(Mark)について説明する。マークは、図9に示されるように、ClipおよびPlaylistの中のハイライトや特徴的な時間を指定するために設けられている。Clipに付加されるマークは、ClipMark (クリップマーク) と呼ばれる。ClipMarkは、AVストリームの内容に起因する特徴的なシーンを指定する、例えば番組の頭だし点やシーンチェンジ点などである。ClipMarkは、図1の例えば解析部14によって生成される。Playlistを再生する時、そのPlaylistが参照するClipのマークを参照して、使用する事ができる。

【0080】Playlistに付加されるマークは、Playlist Mark (プレイリストマーク) と呼ばれる。Playlist Markは、主にユーザーによってセットされる、例えば、ブックマークやリジューム点などである。ClipまたはPlaylistにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去する事である。従って、マークの設定や削除により、AVストリームは何の変更もされない。

【0081】ClipMarkの別のフォーマットとして、Clip Markが参照するピクチャをAVストリームの中でのアドレススペースで指定するようにしても良い。Clipにマークをセットすることは、マーク点のピクチャを示すアドレスベースの情報をマークリストに追加することにより行われる。また、マークを削除することは、マークリストの

10

20

30

40

50



中から、そのマーク点のピクチャを示すアドレススペースの情報を除去する事である。従って、マークの設定や削除により、AVストリームは何の変更もされない。

【0082】次にサムネイルについて説明する。サムネイルは、Volume、PlayList、およびClipに付加される静止画である。サムネイルには、2つの種類があり、1つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル（不図示）などを操作して見たいものを選択するためのメニュー画面で使われるものである。もう1つは、マークが指しているシーンを表す画像である。

【0083】Volumeと各PlayListは代表画を持つことができるようにする必要がある。Volumeの代表画は、ディスク（記録媒体100、以下、記録媒体100はディスク状のものであるとし、適宜、ディスクと記述する）を記録再生装置1の所定の場所にセットした時に、そのディスクの内容を表す静止画を最初に表示する場合などに用いられることを想定している。PlayListの代表画は、PlayListを選択するメニュー画面において、PlayListの内容を表すための静止画として用いられることを想定している。

【0084】PlayListの代表画として、PlayListの最初の画像をサムネイル（代表画）にすることが考えられるが、必ずしも再生時刻0の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、PlayListのサムネイルとして、任意の画像をユーザが設定できるようにする。以上Volumeを表す代表画としてのサムネイルと、PlayListを表す代表画としてのサムネイルの2種類のサムネイルをメニューサムネイルと称する。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出される必要がある。このため、すべてのメニューサムネイルを1つのファイルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画から抜き出したピクチャである必要はなく、図10に示すように、パーソナルコンピュータやデジタルスチルカメラから取り込まれた画像でもよい。

【0085】一方、ClipとPlayListには、複数個のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることが出来るようにする必要がある。このようなマーク点を表すピクチャをマークサムネイル（Mark Thumbnails）と称する。従って、マークサムネイルの元となる画像は、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。

【0086】図11は、PlayListに付けられるマークと、そのマークサムネイルの関係について示す図であり、図12は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、PlayListの詳細を表す時に、サブメニュー等で使われるため、短いアク

セス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置1がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかっても、問題にはならない。

【0087】また、ボリューム内に存在するファイル数を減らすために、すべてのマークサムネイルは1つのファイルに格納するのがよい。PlayListはメニューサムネイル1つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、PlayList経由で指定する）ため、メニューサムネイルを設ける必要はない。

【0088】図13は、上述したことを考慮した場合のメニューサムネイル、マークサムネイル、PlayList、およびClipの関係について示した図である。メニューサムネイルファイルには、PlayList毎に設けられたメニューサムネイルがファイルされている。メニューサムネイルファイルには、ディスクに記録されているデータの内容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各PlayList毎と各Clip毎に作成されたサムネイルがファイルされている。

【0089】次に、CPI（Characteristic Point Information）について説明する。CPIは、Clipインフォメーションファイルに含まれるデータであり、主に、それはClipへのアクセスポイントのタイムスタンプが与えられた時、Clip AV stream fileの中でデータの読み出しを開始すべきデータアドレスを見つけるために用いられる。本実施の形態では、2種類のCPIを用いる。1つは、EP\_mapであり、もう一つは、TU\_mapである。

【0090】EP\_mapは、エントリポイント（EP）データのリストであり、それはエレメンタリストリームおよびトランスポートストリームから抽出されたものである。これは、AVストリームの中でデコードを開始すべきエントリポイントの場所を見つけるためのアドレス情報を持つ。1つのEPデータは、プレゼンテーションタイムスタンプ（PTS）と、そのPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

【0091】EP\_mapは、主に2つの目的のために使用される。第1に、PlayListの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第2に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタックスを解析することができるとき、EP\_mapが作成され、ディスクに記録される。

【0092】TU\_mapは、デジタルインタフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット（TU）データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータ

10

20

30

40

50

アドレスとの関係を与える。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができないとき、TU\_mapが作成され、ディスクに記録される。

【0093】STCInfoは、MPEG2トランスポートストリームをストアしているAVストリームファイルの中にあるSTCの不連続点情報をストアする。

【0094】AVストリームがSTCの不連続点を持つ場合、そのAVストリームファイルの中で同じ値のPTSが現れるかもしれない。そのため、AVストリーム上のある時刻をPTSベースで指す場合、アクセスポイントのPTSだけではそのポイントを特定するためには不十分である。更に、そのPTSを含むところの連続なSTC区間のインデックスが必要である。連続なSTC区間を、このフォーマットではSTC-sequenceと呼び、そのインデックスをSTC-sequence-idと呼ぶ。STC-sequenceの情報は、Clip Information fileのSTCInfoで定義される。

【0095】STC-sequence-idは、EP\_mapを持つAVストリームファイルで使用するものであり、TU\_mapを持つAVストリームファイルではオプションである。

【0096】プログラムは、エレメンタリストリームの集まりであり、これらのストリームの同期再生のために、ただ1つのシステムタイムベースを共有するものである。

【0097】再生装置（図1の記録再生装置1）にとって、AVストリームのデコードに先だち、そのAVストリームの内容がわかることは有用である。例えば、ビデオやオーディオのエレメンタリストリームを伝送するトランスポートパケットのPIDの値や、ビデオやオーディオのコンポーネント種類（例えば、HDTVのビデオとMPEG-2 AACのオーディオストリームなど）などの情報である。この情報はAVストリームを参照するところのPlayListの内容をユーザに説明するメニュー画面を作成するのに有用であるし、また、AVストリームのデコードに先だち、再生装置のAVデコーダおよびデマルチプレクサの初期状態をセットするために役立つ。

【0098】この理由のために、Clip Information fileは、プログラムの内容を説明するためのProgramInfoを持つ。

【0099】MPEG2トランスポートストリームをストアしているAVストリームファイルは、ファイルの中でプログラム内容が変化することもある。例えば、ビデオエレメンタリストリームを伝送するところのトランスポートパケットのPIDが変化したり、ビデオストリームのコンポーネント種類がSDTVからHDTVに変化するなどである。

【0100】ProgramInfoは、AVストリームファイルの中でのプログラム内容の変化点の情報をストアする。AVストリームファイルの中で、このフォーマットで定めるところのプログラム内容が一定である区間をProgram-se-

quenceと呼ぶ。

【0101】Program-sequenceは、EP\_mapを持つAVストリームファイルで使用するものであり、TU\_mapを持つAVストリームファイルではオプションである。

【0102】本実施の形態では、セルフエンコードのストリームフォーマット（SESF）を定義する。SESFは、アナログ入力信号を符号化する目的、およびデジタル入力信号（例えばDV）をデコードしてからMPEG2トランスポートストリームに符号化する場合に用いられる。

【0103】SESFは、MPEG-2トランスポートストリームおよびAVストリームについてのエレメンタリストリームの符号化制限を定義する。記録再生装置1が、SESFストリームをエンコードし、記録する場合、EP\_mapが作成され、ディスクに記録される。

【0104】デジタル放送のストリームは、次に示す方式のうちのいずれかが用いられて記録媒体100に記録される。まず、デジタル放送のストリームをSESFストリームにトランスコーディングする。この場合、記録されたストリームは、SESFに準拠しなければならない。この場合、EP\_mapが作成されて、ディスクに記録されなければならない。

【0105】あるいは、デジタル放送ストリームを構成するエレメンタリストリームを新しいエレメンタリストリームにトランスコーディングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP\_mapが作成されて、ディスクに記録されなければならない。

【0106】例えば、入力ストリームがISDB（日本のデジタルBS放送の規格名称）準拠のMPEG-2トランスポートストリームであり、それがHDTVビデオストリームとMPEG AACオーディオストリームを含むとする。HDTVビデオストリームをSDTVビデオストリームにトランスコーディングし、そのSDTVビデオストリームとオリジナルのAACオーディオストリームをTSに再多重化する。SDTVストリームと記録されるトランスポートストリームは、共にISDBフォーマットに準拠しなければならない。

【0107】デジタル放送のストリームが、記録媒体100に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、その時にEP\_mapが作成されてディスクに記録される。

【0108】または、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、その時にTU\_mapが作成されてディスクに記録される。

【0109】次にディレクトリとファイルについて説明する。以下、記録再生装置1をDVR（Digital Video Recording）と適宜記述する。図14はディスク上のディレ

クトリ構造の一例を示す図である。DVRのディスク上に必要なディレクトリは、図14に示したように、“DVR”ディレクトリを含むrootディレクトリ、“PLAYLIST”ディレクトリ、“CLIPINF”ディレクトリ、“M2TS”ディレクトリ、および“DATA”ディレクトリを含む“DVR”ディレクトリである。rootディレクトリの下に、これら以外のディレクトリを作成されるようにしても良いが、それらは、本実施の形態のアプリケーションフォーマットでは、無視されるとする。

【0110】“DVR”ディレクトリの下には、DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。“DVR”ディレクトリは、4個のディレクトリを含む。“PLAYLIST”ディレクトリの下には、Real PlaylistとVirtual Playlistのデータベースファイルが置かれる。このディレクトリは、Playlistが1つもなくても存在する。

【0111】“CLIPINF”ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが1つもなくても存在する。“M2TS”ディレクトリの下には、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが1つもなくても存在する。“DATA”ディレクトリは、デジタルTV放送などのデータ放送のファイルがストアされる。

【0112】“DVR”ディレクトリは、次に示すファイルをストアする。“info.dvr”ファイルは、DVRディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。DVRディレクトリの下には、ただ一つのinfo.dvrがなければならない。ファイル名は、info.dvrに固定されるとする。“menu.thmb”ファイルは、メニューサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは1つのメニューサムネイルがなければならない。ファイル名は、menu.thmbに固定されるとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくても良い。

【0113】“mark.thmb”ファイルは、マークサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは1つのマークサムネイルがなければならない。ファイル名は、mark.thmbに固定されるとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくても良い。

【0114】“PLAYLIST”ディレクトリは、2種類のPlaylistファイルをストアするものであり、それらは、Real PlaylistとVirtual Playlistである。“xxxxx.rpls”ファイルは、1つのReal Playlistに関連する情報をストアする。それぞれのReal Playlist毎に、1つのファイルが作られる。ファイル名は、“xxxxx.rpls”である。ここで、“xxxxx”は、5個の0乃至9まで数字である。ファイル拡張子は、“rpls”でなければならないとする。

【0115】“yyyyy.vpls”ファイルは、1つのVirtual Playlistに関連する情報をストアする。それぞれのVirt

ual Playlist毎に、1つのファイルが作られる。ファイル名は、“yyyyy.vpls”である。ここで、“yyyyy”は、5個の0乃至9まで数字である。ファイル拡張子は、“vpls”でなければならないとする。

【0116】“CLIPINF”ディレクトリは、それぞれのAVストリームファイルに対応して、1つのファイルをストアする。“zzzzz.clpi”ファイルは、1つのAVストリームファイル(Clip AV stream file または Bridge-Clip AV stream file)に対応するClip Information fileである。ファイル名は、“zzzzz.clpi”であり、“zzzzz”は、5個の0乃至9までの数字である。ファイル拡張子は、“clpi”でなければならないとする。

【0117】“M2TS”ディレクトリは、AVストリームのファイルをストアする。“zzzzz.m2ts”ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは、Clip AV stream fileまたはBridge-Clip AV streamである。ファイル名は、“zzzzz.m2ts”であり、“zzzzz”は、5個の0乃至9までの数字である。ファイル拡張子は、“m2ts”でなければならないとする。

【0118】“DATA”ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMHEGファイルなどである。

【0119】次に、各ディレクトリ(ファイル)のシンタクスとセマンティクスを説明する。まず、“info.dvr”ファイルについて説明する。図15は、“info.dvr”ファイルのシンタクスを示す図である。“info.dvr”ファイルは、3個のオブジェクトから構成され、それらは、DVRVolume()、TableOfPlayLists()、およびMakersPrivateData()である。

【0120】図15に示したinfo.dvrのシンタクスについて説明するに、TableOfPlayLists\_Start\_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlayList()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0121】MakersPrivateData\_Start\_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。padding\_word(パディングワード)は、info.dvrのシンタクスに従って挿入される。N1とN2は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしても良い。

【0122】DVRVolume()は、ボリューム(ディスク)の内容を記述する情報をストアする。図16は、DVRVolume()のシンタクスを示す図である。図16に示したDVRVolume()のシンタクスを説明するに、version\_numberは、このDVRVolume()のバージョンナンバを示す4個のキャラクター文字を示す。version\_numberは、ISO 646に従って、“0045”と符号化される。

【0123】lengthは、このlengthフィールドの直後か

らDVRVolume()の最後までDVRVolume()のバイト数を示す32ビットの符号なし整数で表される。

【0124】ResumeVolume()は、ボリュームの中で最後に再生したReal PlaylistまたはVirtual Playlistのファイル名を記憶している。ただし、Real PlaylistまたはVirtual Playlistの再生をユーザが中断した時の再生位置は、PlaylistMark()において定義されるresume-markにストアされる(図42、図43)。

【0125】図17は、ResumeVolume()のシンタクスを示す図である。図17に示したResumeVolume()のシンタクスを説明するに、valid\_flagは、この1ビットのフラグが1にセットされている場合、resume\_playlist\_nameフィールドが有効であることを示し、このフラグが0にセットされている場合、resume\_playlist\_nameフィールドが無効であることを示す。

【0126】resume\_playlist\_nameの10バイトのフィールドは、リジュームされるべきReal PlaylistまたはVirtual Playlistのファイル名を示す。

【0127】図16に示したDVRVolume()のシンタクスのなかの、UIAppInfoVolumeは、ボリュームについてのユーザインターフェースアプリケーションのパラメータをストアする。図18は、UIAppInfoVolumeのシンタクスを示す図であり、そのセマンティクスを説明するに、character\_setの8ビットのフィールドは、Volume\_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

【0128】name\_lengthの8ビットフィールドは、Volume\_nameフィールドの中に示されるボリューム名のバイト長を示す。Volume\_nameのフィールドは、ボリュームの名称を示す。このフィールドの中の左からname\_length数のバイト数が、有効なキャラクター文字であり、それはボリュームの名称を示す。Volume\_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0129】Volume\_protect\_flagは、ボリュームの中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが1にセットされている場合、ユーザが正しくPIN番号(パスワード)を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せる事(再生される事)が許可される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、そのボリュームのコンテンツを、ユーザに見せる事が許可される。

【0130】最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが0にセットされているか、または、このフラグが1にセットされていてもユーザがPIN番号を正しく入力できたならば、記録再生装置1は、そのディスクの中のPlaylistの一覧を表示させる。それぞれのPlaylistの再生制限は、volume\_p

protect\_flagとは無関係であり、それはUIAppInfoPlaylist()の中に定義されるplayback\_control\_flagによって示される。

【0131】PINは、4個の0乃至9までの数字で構成され、それぞれの数字は、ISO/IEC 646に従って符号化される。ref\_thumbnail\_indexのフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。ref\_thumbnail\_indexフィールドが、0xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref\_thumbnail\_indexの値を用いて参照される。ref\_thumbnail\_indexフィールドが、0xFFFFである場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

【0132】次に図15に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説明する。TableOfPlayLists()は、Playlist(Real PlaylistとVirtual Playlist)のファイル名をストアする。ボリュームに記録されているすべてのPlaylistファイルは、TableOfPlayLists()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlaylistのデフォルトの再生順序を示す。

【0133】図20は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明するに、TableOfPlayListsのversion\_numberは、このTableOfPlayListsのバージョンナンバーを示す4個のキャラクター文字を示す。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0134】lengthは、このlengthフィールドの直後からTableOfPlayLists()の最後までTableOfPlayLists()のバイト数を示す32ビットの符号なしの整数である。number\_of\_PlayListsの16ビットのフィールドは、Playlist\_file\_nameを含むfor-loopのループ回数を示す。この数字は、ボリュームに記録されているPlaylistの数に等しくなければならない。Playlist\_file\_nameの10バイトの数字は、Playlistのファイル名を示す。

【0135】図21は、TableOfPlayLists()のシンタクスの別の構成を示す図である。図21に示したシンタクスは、図20に示したシンタクスに、UIAppInfoPlaylist(後述)を含ませた構成とされている。このように、UIAppInfoPlaylistを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図20に示したシンタクスを用いるとして以下の説明をする。

【0136】図15に示したinfo.dvrのシンタクス内のMakersPrivateDataについて説明する。MakersPrivateDataは、記録再生装置1のメーカーが、各社の特別なアプリケーションのために、MakersPrivateData()の中にメーカーのプライベートデータを挿入できるように設けられている。各メーカーのプライベートデータは、それを定義し

10

20

30

40

50

たメーカを識別するために標準化されたmaker\_IDを持つ。MakersPrivateData()は、1つ以上のmaker\_IDを含んでも良い。

【0137】所定のメーカが、プライベートデータを挿入したい時に、すでに他のメーカのプライベートデータがMakersPrivateData()に含まれていた場合、他のメーカは、既にある古いプライベートデータを消去するのではなく、新しいプライベートデータをMakersPrivateData()の中に追加するようにする。このように、本実施の形態においては、複数のメーカのプライベートデータが、1つのMakersPrivateData()に含まれることが可能であるようにする。

【0138】図22は、MakersPrivateDataのシンタクスを示す図である。図22に示したMakersPrivateDataのシンタクスについて説明するに、version\_numberは、このMakersPrivateData()のバージョンナンバを示す4個のキャラクター文字を示す。version\_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からMakersPrivateData()の最後までMakersPrivateData()のバイト数を示す32ビットの符号なし整数を示す。

【0139】mpd\_blocks\_start\_addressは、MakersPrivateData()の先頭のバイトからの相対バイト数を単位として、最初のmpd\_block()の先頭バイトアドレスを示す。相対バイト数はゼロからカウントされる。number\_of\_maker\_entriesは、MakersPrivateData()の中に含まれているメーカプライベートデータのエン트리数を与える16ビットの符号なし整数である。MakersPrivateData()の中に、同じmaker\_IDの値を持つメーカプライベートデータが2個以上存在してはならない。

【0140】mpd\_block\_sizeは、1024バイトを単位として、1つのmpd\_blockの大きさを与える16ビットの符号なし整数である。例えば、mpd\_block\_size=1ならば、それは1つのmpd\_blockの大きさが1024バイトであることを示す。number\_of\_mpd\_blocksは、MakersPrivateData()の中に含まれるmpd\_blockの数を与える16ビットの符号なし整数である。maker\_IDは、そのメーカプライベートデータを作成したDVRシステムの製造メーカを示す16ビットの符号なし整数である。maker\_IDに符号化される値は、このDVRフォーマットのライセンスによって指定される。

【0141】maker\_model\_codeは、そのメーカプライベートデータを作成したDVRシステムのモデルナンバーコードを示す16ビットの符号なし整数である。maker\_model\_codeに符号化される値は、このフォーマットのライセンスを受けた製造メーカによって設定される。start\_mpd\_block\_numberは、そのメーカプライベートデータが開始されるmpd\_blockの番号を示す16ビットの符号なし整数である。メーカプライベートデータの先頭データは、mpd\_blockの先頭にアラインされなければならない

い。start\_mpd\_block\_numberは、mpd\_blockのfor-loopの中の変数jに対応する。

【0142】mpd\_lengthは、バイト単位でメーカプライベートデータの大きさを示す32ビットの符号なし整数である。mpd\_blockは、メーカプライベートデータがストアされる領域である。MakersPrivateData()の中のすべてのmpd\_blockは、同じサイズでなければならない。

【0143】次に、Real Playlist fileとVirtual Playlist fileについて、換言すれば、xxxxx.rplsとyyyyy.vplsについて説明する。図23は、xxxxx.rpls (Real Playlist)、または、yyyyy.vpls (Virtual Playlist)のシンタクスを示す図である。xxxxx.rplsとyyyyy.vplsは、同一のシンタクス構成をもつ。xxxxx.rplsとyyyyy.vplsは、それぞれ、3個のオブジェクトから構成され、それらは、PlayList()、PlayListMark()、およびMakersPrivateData()である。

【0144】PlayListMark\_Start\_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayListMark()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0145】MakersPrivateData\_Start\_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0146】padding\_word (パディングワード) は、PlayListファイルのシンタクスにしたがって挿入され、N1とN2は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしても良い。

【0147】ここで、既に、簡便に説明したが、PlayListについてさらに説明する。ディスク内にあるすべてのReal Playlistによって、Bridge-Clip (後述)を除くすべてのClipの中の再生区間が参照されていなければならない。かつ、2つ以上のReal Playlistが、それらのPlayItemで示される再生区間を同一のClipの中でオーバーラップさせてはならない。

【0148】図24を参照してさらに説明するに、図24(A)に示したように、全てのClipは、対応するReal Playlistが存在する。この規則は、図24(B)に示したように、編集作業が行われた後においても守られる。従って、全てのClipは、どれかしのReal Playlistを参照することにより、必ず視聴することが可能である。

【0149】図24(C)に示したように、Virtual Playlistの再生区間は、Real Playlistの再生区間またはBridge-Clipの再生区間の中に含まれていなければならない。どのVirtual Playlistにも参照されないBridge-Clipがディスクの中に存在してはならない。

【0150】Real Playlistは、PlayItemのリストを含むが、SubPlayItemを含んではならない。Virtual Playl

istは、PlayItemのリストを含み、PlayList()の中に示されるCPI\_typeがEP\_map typeであり、かつPlayList\_typeが0 (ビデオとオーディオを含むPlayList) である場合、Virtual PlayListは、ひとつのSubPlayItemを含む事ができる。本実施の形態におけるPlayList()では、SubPlayItemはオーディオのアフレコの目的にだけに使用される、そして、1つのVirtual PlayListが持つSubPlayItemの数は、0または1でなければならない。

【0151】次に、PlayListについて説明する。図25は、PlayListのシンタクスを示す図である。図25に示したPlayListのシンタクスを説明するに、version\_numberは、このPlayList()のバージョンナンバーを示す4個のキャラクター文字である。version\_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からPlayList()の最後までPlayList()のバイト数を示す32ビットの符号なし整数である。PlayList\_typeは、このPlayListのタイプを示す8ビットのフィールドであり、その一例を図26に示す。

【0152】CPI\_typeは、1ビットのフラグであり、PlayItem()およびSubPlayItem()によって参照されるClipのCPI\_typeの値を示す。1つのPlayListによって参照される全てのClipは、それらのCPI()の中に定義されるCPI\_typeの値が同じでなければならない。number\_of\_PlayItemsは、PlayListの中にあるPlayItemの数を示す16ビットのフィールドである。

【0153】所定のPlayItem()に対応するPlayItem\_idは、PlayItem()を含むfor-loopの中で、そのPlayItem()の現れる順番により定義される。PlayItem\_idは、0から開始される。number\_of\_SubPlayItemsは、PlayListの中にあるSubPlayItemの数を示す16ビットのフィールドである。この値は、0または1である。付加的なオーディオストリームのパス(オーディオストリームパス)は、サブパスの一種である。

【0154】次に、図25に示したPlayListのシンタクスのUIAppInfoPlayListについて説明する。UIAppInfoPlayListは、PlayListについてのユーザインターフェースアプリケーションのパラメータをストアする。図27は、UIAppInfoPlayListのシンタクスを示す図である。図27に示したUIAppInfoPlayListのシンタクスを説明するに、character\_setは、8ビットのフィールドであり、PlayList\_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示したテーブルに準拠する値に対応する。

【0155】name\_lengthは、8ビットフィールドであり、PlayList\_nameフィールドの中に示されるPlayList名のバイト長を示す。PlayList\_nameのフィールドは、PlayListの名称を示す。このフィールドの中の左からname\_length数のバイト数が、有効なキャラクター文字であり、それはPlayListの名称を示す。PlayList\_nameフィ

ールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0156】record\_time\_and\_dateは、PlayListが記録された時の日時をストアする56ビットのフィールドである。このフィールドは、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、2001/12/23:01:02:03 は、“0x20011223010203”と符号化される。

【0157】durationは、PlayListの総再生時間を時間/分/秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、01:45:30は、“0x014530”と符号化される。

【0158】valid\_periodは、PlayListが有効である期間を示す32ビットのフィールドである。このフィールドは、8個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、記録再生装置1は、この有効期間の過ぎたPlayListを自動消去する、といったように用いられる。例えば、2001/05/07 は、“0x20010507”と符号化される。

【0159】maker\_idは、そのPlayListを最後に更新したDVRプレーヤ(記録再生装置1)の製造者を示す16ビットの符号なし整数である。maker\_idに符号化される値は、DVRフォーマットのライセンスによって割り当てられる。maker\_codeは、そのPlayListを最後に更新したDVRプレーヤのモデル番号を示す16ビットの符号なし整数である。maker\_codeに符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

【0160】playback\_control\_flagのフラグが1にセットされている場合、ユーザが正しくPIN番号を入力してきた場合にだけ、そのPlayListは再生される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、そのPlayListを視聴することができる。

【0161】write\_protect\_flagは、図28(A)にテーブルを示すように、1にセットされている場合、write\_protect\_flagを除いて、そのPlayListの内容は、消去および変更されない。このフラグが0にセットされている場合、ユーザは、そのPlayListを自由に消去および変更できる。このフラグが1にセットされている場合、ユーザが、そのPlayListを消去、編集、または上書きする前に、記録再生装置1はユーザに再確認するようなメッセージを表示させる。

【0162】write\_protect\_flagが0にセットされているReal PlayListが存在し、かつ、そのReal PlayListのClipを参照するVirtual PlayListが存在し、そのVirtual PlayListのwrite\_protect\_flagが1にセットされていても良い。ユーザが、Real PlayListを消去しようとする場合、記録再生装置1は、そのReal PlayListを消去す

る前に、上記Virtual Playlistの存在をユーザに警告するか、または、そのReal Playlistを"Minimize"する。

【0163】is\_played\_flagは、図28(B)に示すように、フラグが1にセットされている場合、そのPlaylistは、記録されてから一度は再生されたことを示し、0にセットされている場合、そのPlaylistは、記録されてから一度も再生されたことがないことを示す。

【0164】archiveは、図28(C)に示すように、そのPlaylistがオリジナルであるか、コピーされたものであるかを示す2ビットのフィールドである。ref\_thumbnail\_indexのフィールドは、Playlistを代表するサムネイル画像の情報を示す。ref\_thumbnail\_indexフィールドが、0xFFFFでない値の場合、そのPlaylistには、Playlistを代表するサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref\_thumbnail\_indexの値を用いて参照される。ref\_thumbnail\_indexフィールドが、0xFFFFである場合、そのPlaylistには、Playlistを代表するサムネイル画像が付加されていない。

【0165】次にPlayItemについて説明する。1つのPlayItem()は、基本的に次のデータを含む。Clipのファイル名を指定するためのClip\_information\_file\_name、Clipの再生区間を特定するためのIN\_timeとOUT\_timeのペア、Playlist()において定義されるCPI\_typeがEP\_map typeである場合、IN\_timeとOUT\_timeが参照するところのSTC\_sequence\_id、および、先行するPlayItemと現在のPlayItemとの接続の状態を示すところのconnection\_conditionである。

【0166】Playlistが2つ以上のPlayItemから構成される時、それらのPlayItemはPlaylistのグローバル時間軸上に、時間のギャップまたはオーバーラップなしに一列に並べられる。Playlist()において定義されるCPI\_typeがEP\_map typeであり、かつ現在のPlayItemがBridgeSequence()を持たない時、そのPlayItemにおいて定義されるIN\_timeとOUT\_timeのペアは、STC\_sequence\_idによって指定される同じSTC連続区間上の時間を指していないなければならない。そのような例を図29に示す。

【0167】図30は、Playlist()において定義されるCPI\_typeがEP\_map typeであり、かつ現在のPlayItemがBridgeSequence()を持つ時、次に説明する規則が適用される場合を示している。現在のPlayItemに先行するPlayItemのIN\_time(図の中でIN\_time1と示されているもの)は、先行するPlayItemのSTC\_sequence\_idによって指定されるSTC連続区間上の時間を指している。先行するPlayItemのOUT\_time(図の中でOUT\_time1と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このOUT\_timeは、後述する符号化制限に従っていないなければならない。

【0168】現在のPlayItemのIN\_time(図の中でIN\_time2と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN\_timeも、後述する符号化制限に従っていないなければならない。現在のPlayItemのPlayItemのOUT\_time(図の中でOUT\_time2と示されているもの)は、現在のPlayItemのSTC\_sequence\_idによって指定されるSTC連続区間上の時間を指している。

【0169】図31に示すように、Playlist()のCPI\_typeがTU\_map typeである場合、PlayItemのIN\_timeとOUT\_timeのペアは、同じClip AVストリーム上の時間を指している。

【0170】PlayItemのシンタクスは、図32に示すようになる。図32に示したPlayItemのシンタクスを説明するに、Clip\_Information\_file\_nameのフィールドは、ClipInformation fileのファイル名を示す。このClipInformation fileのClipInfo()において定義されるClip\_stream\_typeは、Clip AV streamを示していなければならない。

【0171】STC\_sequence\_idは、8ビットのフィールドであり、PlayItemが参照するSTC連続区間のSTC\_sequence\_idを示す。Playlist()の中で指定されるCPI\_typeがTU\_map typeである場合、この8ビットフィールドは何も意味を持たず、0にセットされる。IN\_timeは、32ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN\_timeのセマンティクスは、図33に示すように、Playlist()において定義されるCPI\_typeによって異なる。

【0172】OUT\_timeは、32ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT\_timeのセマンティクスは、図34に示すように、Playlist()において定義されるCPI\_typeによって異なる。

【0173】Connection\_Conditionは、図35に示したような先行するPlayItemと、現在のPlayItemとの間の接続状態を示す2ビットのフィールドである。図36は、図35に示したConnection\_Conditionの各状態について説明する図である。

【0174】次に、BridgeSequenceInfo()について、図37を参照して説明する。BridgeSequenceInfo()は、現在のPlayItemの付属情報であり、次に示す情報を持つ。Bridge-Clip AV streamファイルとそれに対応するClip Information file(図45)を指定するBridge\_Clip\_Information\_file\_nameを含む。

【0175】また、先行するPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。このアドレスは、RSPN\_exit\_from\_previous\_Clipと称される。さらに現在のPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットの前にBr

idge-Clip AV streamファイルの最後のソースパケットが接続される。このアドレスは、RSPN\_enter\_to\_current\_Clipと称される。

【0176】図37において、RSPN\_arrival\_time\_discontinuityは、the Bridge-Clip AVstreamファイルの中でアライバルタイムベースの不連続点があるところのソースパケットのアドレスを示す。このアドレスは、Clip Info() (図46) の中において定義される。

【0177】図38は、BridgeSequenceinfoのシンタクスを示す図である。図38に示したBridgeSequenceinfoのシンタクスを説明するに、Bridge\_Clip\_Information\_10 file\_nameのフィールドは、Bridge-Clip AV streamファイルに対応するClip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip\_stream\_typeは、'Bridge-Clip AV stream'を示していなければならない。

【0178】RSPN\_exit\_from\_previous\_Clipの32ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。RSPN\_exit\_from\_previous\_Clipは、ソースパケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset\_SPNの値を初期値としてカウントされる。

【0179】RSPN\_enter\_to\_current\_Clipの32ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN\_exit\_from\_previous\_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset\_SPNの値を初期値としてカウントされる。

【0180】次に、SubPlayItemについて、図39を参照して説明する。SubPlayItem()の使用は、PlayList()のCPI\_typeがEP\_map typeである場合に許される。本実施の形態においては、SubPlayItemはオーディオのアフレコの目的のためだけに使用されるとする。SubPlayItem()は、次に示すデータを含む。まず、PlayListの中のsub\_pathが参照するClipを指定するためのClip\_information\_file\_nameを含む。

【0181】また、Clipの中のsub\_pathの再生区間を指定するためのSubPath\_IN\_time と SubPath\_OUT\_timeを含む。さらに、main pathの時間軸上でsub\_pathが再生開始する時刻を指定するためのsync\_PlayItem\_id と sync\_start\_PTS\_of\_PlayItemを含む。sub\_pathに参照されるオーディオのClip AV streamは、STC不連続点 (シス

テムタイムベースの不連続点) を含んではならない。sub\_pathに使われるClipのオーディオサンプルのクロックは、main pathのオーディオサンプルのクロックにロックされている。

【0182】図40は、SubPlayItemのシンタクスを示す図である。図40に示したSubPlayItemのシンタクスを説明するに、Clip\_Information\_file\_nameのフィールドは、Clip Information fileのファイル名を示し、それはPlayListの中でsub\_pathによって使用される。このClip Information fileのClipInfo()において定義されるClip\_stream\_typeは、Clip AV streamを示していなければならない。

【0183】SubPath\_typeの8ビットのフィールドは、sub\_pathのタイプを示す。ここでは、図41に示すように、'0x00'しか設定されておらず、他の値は、将来のために確保されている。

【0184】sync\_PlayItem\_idの8ビットのフィールドは、main pathの時間軸上でsub\_pathが再生開始する時刻が含まれるPlayItemのPlayItem\_idを示す。所定のPlayItemに対応するPlayItem\_idの値は、PlayList()において定義される (図25参照)。

【0185】sync\_start\_PTS\_of\_PlayItemの32ビットのフィールドは、main pathの時間軸上でsub\_pathが再生開始する時刻を示し、sync\_PlayItem\_idで参照されるPlayItem上のPTS(Presentation Time Stamp)の上位32ビットを示す。SubPath\_IN\_timeの32ビットフィールドは、Sub\_pathの再生開始時刻をストアする。SubPath\_IN\_timeは、Sub\_Pathの中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示す。

【0186】SubPath\_OUT\_timeの32ビットフィールドは、Sub\_pathの再生終了時刻をストアする。SubPath\_OUT\_timeは、次式によって算出されるPresentation\_end\_TSの値の上位32ビットを示す。

$$\text{Presentation\_end\_TS} = \text{PTS\_out} + \text{AU\_duration}$$

ここで、PTS\_outは、SubPathの最後のプレゼンテーションユニットに対応する33ビット長のPTSである。AU\_durationは、SubPathの最後のプレゼンテーションユニットの90kHz単位の表示期間である。

【0187】次に、図23に示したxxxxx.rplsとyyyyy.vplsのシンタクス内のPlayListMark()について説明する。PlayListについてのマーク情報は、このPlayListMarkにストアされる。図42は、PlayListMarkのシンタクスについて説明するに、version\_numberは、このPlayListMark()のバージョンナンバを示す4個のキャラクター文字である。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0188】lengthは、このlengthフィールドの直後からPlayListMark()の最後までのPlayListMark()のバイト



数を示す32ビットの符号なし整数である。number\_of\_Playlist\_marksは、PlaylistMarkの中にストアされているマークの個数を示す16ビットの符号なし整数である。number\_of\_Playlist\_marksは、0であってもよい。mark\_typeは、マークのタイプを示す8ビットのフィールドであり、図43に示すテーブルに従って符号化される。

【0189】mark\_time\_stampの32ビットフィールドは、マークが指定されたポイントを示すタイムスタンプをストアする。mark\_time\_stampのセマンティクスは、図44に示すように、Playlist()において定義されるCPI\_typeによって異なる。PlayItem\_idは、マークが置かれているところのPlayItemを指定する8ビットのフィールドである。所定のPlayItemに対応するPlayItem\_idの値は、Playlist()において定義される(図25参照)。

【0190】character\_setの8ビットのフィールドは、mark\_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示した値に対応する。name\_lengthの8ビットフィールドは、Mark\_nameフィールドの中に示されるマーク名のバイト長を示す。mark\_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname\_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。Mark\_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どのような値が設定されても良い。

【0191】ref\_thumbnail\_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref\_thumbnail\_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref\_thumbnail\_indexの値を用いて参照される(後述)。ref\_thumbnail\_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない事を示す。

【0192】次に、Clip information fileについて説明する。zzzzz.clpi (Clip information fileファイル)は、図45に示すように6個のオブジェクトから構成される。それらは、ClipInfo()、STC\_Info()、ProgramInfo()、CPI()、ClipMark()、およびMakersPrivateData()である。AVストリーム(Clip AVストリームまたはBridge-Clip AV stream)とそれに対応するClip Informationファイルは、同じ数字列の"zzzzz"が使用される。

【0193】図45に示したzzzzz.clpi (Clip information fileファイル)のシンタクスについて説明するに、ClipInfo\_Start\_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0194】STC\_Info\_Start\_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、STC\_Info()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。ProgramInfo\_Start\_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ProgramInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。CPI\_Start\_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、CPI()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0195】ClipMark\_Start\_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipMark()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。MakersPrivateData\_Start\_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。padding\_word (パディングワード)は、zzzzz.clpiファイルのシンタクスにしたがって挿入される。N1, N2, N3, N4, およびN5は、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしても良い。

【0196】次に、ClipInfoについて説明する。図46は、ClipInfoのシンタクスを示す図である。ClipInfo()は、それに対応するAVストリームファイル(Clip AVストリームまたはBridge-Clip AVストリームファイル)の属性情報をストアする。

【0197】図46に示したClipInfoのシンタクスについて説明するに、version\_numberは、このClipInfo()のバージョンナンバーを示す4個のキャラクター文字である。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo()の最後までClipInfo()のバイト数を示す32ビットの符号なし整数である。Clip\_stream\_typeの8ビットのフィールドは、図47に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

【0198】offset\_SPNの32ビットのフィールドは、AVストリーム(Clip AVストリームまたはBridge-Clip AVストリーム)ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初にディスクに記録される時、このoffset\_SPNは0でなければならない。

【0199】図48に示すように、AVストリームファイルのはじめの部分が編集によって消去された時、offset\_SPNは、ゼロ以外の値をとっても良い。本実施の形態では、offset\_SPNを参照する相対ソースパケット番号(相対アドレス)が、しばしば、RSPN\_xxx (xxxは変形する。例、RSPN\_EP\_start)の形式でシンタクスの中に記

10

20

30

40

50

述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからoffset\_SPNの値を初期値としてカウントされる。

【0200】AVストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数 (SPN<sub>xxx</sub>) は、次式で算出される。

$$SPN_{xxx} = RSPN_{xxx} - offset\_SPN$$

図48に、offset\_SPN が、4である場合の例を示す。

【0201】TS\_recording\_rateは、24ビットの符号なし整数であり、この値は、DVRドライブ（書き込み部22）へまたはDVRドライブ（読み出し部28）からのAVストリームの必要な入出力のビットレートを与える。record\_time\_and\_dateは、Clipに対応するAVストリームが記録された時の日時をストアする56ビットのフィールドであり、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、2001/12/23:01:02:03は、"0x20011223010203"と符号化される。

【0202】durationは、Clipの総再生時間をリアルタイムクロックに基づいた時間/分/秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、01:45:30は、"0x014530"と符号化される。

【0203】time\_controlled\_flagのフラグは、AVストリームファイルの記録モードを示す。このtime\_controlled\_flagが1である場合、記録モードは、記録してからの時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

$$TS\_average\_rate \cdot 192/188 \cdot (t - start\_time) - \alpha \leq size\_clip(t)$$

$$\leq TS\_average\_rate \cdot 192/188 \cdot (t - start\_time) + \alpha$$

ここで、TS\_average\_rateは、AVストリームファイルのトランスポートストリームの平均ビットレートをbytes/second の単位で表したものである。

【0204】また、上式において、t は、秒単位で表される時間を示し、start\_timeは、AVストリームファイルの最初のソースパケットが記録された時の時刻であり、秒単位で表される。size\_clip(t)は、時刻tにおけるAVストリームファイルのサイズをバイト単位で表したものであり、例えば、start\_timeから時刻tまでに10個のソースパケットが記録された場合、size\_clip(t)は10・192バイトである。αは、TS\_average\_rateに依存する定数である。

【0205】time\_controlled\_flagが0にセットされている場合、記録モードは、記録の時間経過とAVストリームのファイルサイズが比例するように制御していないこ

とを示す。例えば、これは入力トランスポートストリームをトランスペアレント記録する場合である。

【0206】TS\_average\_rateは、time\_controlled\_flagが1にセットされている場合、この24ビットのフィールドは、上式で用いているTS\_average\_rateの値を示す。time\_controlled\_flagが0にセットされている場合、このフィールドは、何も意味を持たず、0にセットされなければならない。例えば、可変ビットレートのトランスポートストリームは、次に示す手順により符号化される。まずトランスポートレートをTS\_recording\_rateの値にセットする。次に、ビデオストリームを可変ビットレートで符号化する。そして、ヌルパケットを使用しない事によって、間欠的にトランスポートパケットを符号化する。

【0207】RSPN\_arrival\_time\_discontinuityの32ビットフィールドは、Bridge-Clip AV streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。RSPN\_arrival\_time\_discontinuityは、ソースパケット番号を単位とする大きさであり、Bridge-Clip AV streamファイルの最初のソースパケットからClipInfo() において定義されるoffset\_SPNの値を初期値としてカウントされる。そのBridge-Clip AV streamファイルの中での絶対アドレスは、上述した

$$SPN_{xxx} = RSPN_{xxx} - offset\_SPN$$

に基づいて算出される。

【0208】reserved\_for\_system\_useの144ビットのフィールドは、システム用にリザーブされている。is\_format\_identifier\_validのフラグが1である時、format\_identifierのフィールドが有効であることを示す。is\_original\_network\_ID\_validのフラグが1である場合、original\_network\_IDのフィールドが有効であることを示す。is\_transport\_stream\_ID\_validのフラグが1である場合、transport\_stream\_IDのフィールドが有効であることを示す。is\_service\_ID\_validのフラグが1である場合、service\_IDのフィールドが有効であることを示す。

【0209】is\_country\_code\_validのフラグが1である時、country\_codeのフィールドが有効であることを示す。format\_identifierの32ビットフィールドは、トランスポートストリームの中でregistration\_descriptor (ISO/IEC13818-1で定義されている) が持つformat\_identifierの値を示す。original\_network\_IDの16ビットフィールドは、トランスポートストリームの中で定義されているoriginal\_network\_IDの値を示す。transport\_stream\_IDの16ビットフィールドは、トランスポートストリームの中で定義されているtransport\_stream\_IDの値を示す。

【0210】service\_IDの16ビットフィールドは、トランスポートストリームの中で定義されているservice\_IDの値を示す。country\_codeの24ビットのフィールド

10

20

30

40

50

は、ISO3166によって定義されるカントリコードを示す。それぞれのキャラクター文字は、ISO8859-1で符号化される。例えば、日本は"JPN"と表され、"0x4A 0x500x4E"と符号化される。stream\_format\_nameは、トランスポートストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の16個のキャラクターコードである。このフィールドの中の無効なバイトは、値'0xFF'がセットされる。

【0211】format\_identifier、original\_network\_ID、transport\_stream\_ID、service\_ID、country\_code、およびstream\_format\_nameは、トランスポートストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI(サービスインフォメーション)の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコーダシステムの初期設定を行うために用いることが可能である。

【0212】次に、STC\_Infoについて説明する。ここでは、MPEG-2トランスポートストリームの中でSTCの不連続点(システムタイムベースの不連続点)を含まない時間区間をSTC\_sequenceと称し、Clipの中で、STC\_sequenceは、STC\_sequence\_idの値によって特定される。図50は、連続なSTC区間について説明する図である。同じSTC\_sequenceの中で同じSTCの値は、決して現れない(ただし、後述するように、Clipの最大時間長は制限されている)。従って、同じSTC\_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、N(N>0)個のSTC不連続点を含む場合、Clipのシステムタイムベースは、(N+1)個のSTC\_sequenceに分割される。

【0213】STC\_Infoは、STCの不連続(システムタイムベースの不連続)が発生する場所のアドレスをストアする。図51を参照して説明するように、RSPN\_STC\_startが、そのアドレスを示し、最後のSTC\_sequenceを除くk番目(k>=0)のSTC\_sequenceは、k番目のRSPN\_STC\_startで参照されるソースパケットが到着した時刻から始まり、(k+1)番目のRSPN\_STC\_startで参照されるソースパケットが到着した時刻で終わる。最後のSTC\_sequenceは、最後のRSPN\_STC\_startで参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

【0214】図52は、STC\_Infoのシンタクスを示す図である。図52に示したSTC\_Infoのシンタクスについて説明するに、version\_numberは、このSTC\_Info()のバージョンナンバーを示す4個のキャラクター文字である。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0215】lengthは、このlengthフィールドの直後か

らSTC\_Info()の最後までSTC\_Info()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI\_typeがTU\_map typeを示す場合、このlengthフィールドはゼロをセットしても良い。CPI()のCPI\_typeがEP\_map typeを示す場合、num\_of\_STC\_sequencesは1以上の値でなければならない。

【0216】num\_of\_STC\_sequencesの8ビットの符号なし整数は、Clipの中でのSTC\_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。所定のSTC\_sequenceに対応するSTC\_sequence\_idは、RSPN\_STC\_startを含むfor-loopの中で、そのSTC\_sequenceに対応するRSPN\_STC\_startの現れる順番により定義されるものである。STC\_sequence\_idは、0から開始される。

【0217】RSPN\_STC\_startの32ビットフィールドは、AVストリームファイル上でSTC\_sequenceが開始するアドレスを示す。RSPN\_STC\_startは、AVストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN\_STC\_startは、AVストリームの中で新しいシステムタイムベースの最初のPCRを持つソースパケットの相対アドレスとしても良い。RSPN\_STC\_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset\_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、既に上述した

$$SPN_{xxx} = RSPN_{xxx} - offset\_SPN$$

により算出される。

【0218】次に、図45に示したzzzzz.clipのシンタクス内のProgramInfoについて説明する。図53を参照しながら説明するに、ここでは、Clipの中で次の特徴をもつ時間区間をprogram\_sequenceと呼ぶ。まず、PCR\_PIDの値が変わらない。次に、ビデオエレメンタリストリームの数が変化しない。また、それぞれのビデオストリームについてのPIDの値とそのVideoCodingInfoによって定義される符号化情報が変化しない。さらに、オーディオエレメンタリストリームの数が変化しない。また、それぞれのオーディオストリームについてのPIDの値とそのAudioCodingInfoによって定義される符号化情報が変化しない。

【0219】program\_sequenceは、同一の時刻において、ただ1つのシステムタイムベースを持つ。program\_sequenceは、同一の時刻において、ただ1つのPMTを持つ。ProgramInfo()は、program\_sequenceが開始する場所のアドレスをストアする。RSPN\_program\_sequence\_startが、そのアドレスを示す。

【0220】図54は、ProgramInfoのシンタクスを示す図である。図54に示したProgramInfoのシンタクスを説明するに、version\_numberは、このProgramInfo()のバージョンナンバーを示す4個のキャラクター文字であ

る。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0221】lengthは、このlengthフィールドの直後からProgramInfo()の最後までProgramInfo()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI\_typeがTU\_map typeを示す場合、このlengthフィールドはゼロにセットされても良い。CPI()のCPI\_typeがEP\_map typeを示す場合、number\_of\_programsは1以上の値でなければならない。

【0222】number\_of\_program\_sequencesの8ビットの符号なし整数は、Clipの中でのprogram\_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。Clipの中でprogram\_sequenceが変化しない場合、number\_of\_program\_sequencesは1をセットされなければならない。RSPN\_program\_sequence\_startの32ビットフィールドは、AVストリームファイル上でプログラムシーケンスが開始する場所の相対アドレスである。

【0223】RSPN\_program\_sequence\_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset\_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

$$\text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset\_SPN}$$

により算出される。シンタクスのfor-loopの中でRSPN\_program\_sequence\_start値は、昇順に現れなければならない。

【0224】PCR\_PIDの16ビットフィールドは、そのprogram\_sequenceに有効なPCRフィールドを含むトランスポートパケットのPIDを示す。number\_of\_videosの8ビットフィールドは、video\_stream\_PIDとVideoCodingInfo()を含むfor-loopのループ回数を示す。number\_of\_audiosの8ビットフィールドは、audio\_stream\_PIDとAudioCodingInfo()を含むfor-loopのループ回数を示す。video\_stream\_PIDの16ビットフィールドは、そのprogram\_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo()は、そのvideo\_stream\_PIDで参照されるビデオストリームの内容を説明しなければならない。

【0225】audio\_stream\_PIDの16ビットフィールドは、そのprogram\_sequenceに有効なオーディオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くAudioCodingInfo()は、そのaudio\_stream\_PIDで参照されるビデオストリームの内容を説明しなければならない。

【0226】なお、シンタクスのfor-loopの中でvideo\_stream\_PIDの値の現れる順番は、そのprogram\_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタク

スのfor-loopの中でaudio\_stream\_PIDの値の現れる順番は、そのprogram\_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

【0227】図55は、図54に示したProgramInfoのシンタクス内のVideoCodingInfoのシンタクスを示す図である。図55に示したVideoCodingInfoのシンタクスを説明するに、video\_formatの8ビットフィールドは、図56に示すように、ProgramInfo()の中のvideo\_stream\_PIDに対応するビデオフォーマットを示す。

【0228】frame\_rateの8ビットフィールドは、図57に示すように、ProgramInfo()の中のvideo\_stream\_PIDに対応するビデオのフレームレートを示す。display\_aspect\_ratioの8ビットフィールドは、図58に示すように、ProgramInfo()の中のvideo\_stream\_PIDに対応するビデオの表示アスペクト比を示す。

【0229】図59は、図54に示したProgramInfoのシンタクス内のAudioCodingInfoのシンタクスを示す図である。図59に示したAudioCodingInfoのシンタクスを説明するに、audio\_codingの8ビットフィールドは、図60に示すように、ProgramInfo()の中のaudio\_stream\_PIDに対応するオーディオの符号化方法を示す。

【0230】audio\_component\_typeの8ビットフィールドは、図61に示すように、ProgramInfo()の中のaudio\_stream\_PIDに対応するオーディオのコンポーネントタイプを示す。sampling\_frequencyの8ビットフィールドは、図62に示すように、ProgramInfo()の中のaudio\_stream\_PIDに対応するオーディオのサンプリング周波数を示す。

【0231】次に、図45に示したzzzzz.clipのシンタクス内のCPI (Characteristic Point Information)について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連づけるためである。CPIには2つのタイプがあり、それらはEP\_mapとTU\_mapである。図63に示すように、CPI()の中のCPI\_typeがEP\_map typeの場合、そのCPI()はEP\_mapを含む。図64に示すように、CPI()の中のCPI\_typeがTU\_map typeの場合、そのCPI()はTU\_mapを含む。1つのAVストリームは、1つのEP\_mapまたは一つのTU\_mapを持つ。AVストリームがSESFトランスポートストリームの場合、それに対応するClipはEP\_mapを持たなければならない。

【0232】図65は、CPIのシンタクスを示す図である。図65に示したCPIのシンタクスを説明するに、version\_numberは、このCPI()のバージョンナンバを示す4個のキャラクター文字である。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後のCPI()のバイト数を示す32ビットの符号なし整数である。CPI\_typeは、図66に示すように、1ビットのフラグであり、ClipのCPIのタイプを表す。

【0233】次に、図65に示したCPIのシンタクス内のEP\_mapについて説明する。EP\_mapには、2つのタイプがあり、それはビデオストリーム用のEP\_mapとオーディオストリーム用のEP\_mapである。EP\_mapの中のEP\_map\_typeが、EP\_mapのタイプを区別する。Clipが1つ以上のビデオストリームを含む場合、ビデオストリーム用のEP\_mapが使用されなければならない。Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、オーディオストリーム用のEP\_mapが使用されなければならない。

【0234】ビデオストリーム用のEP\_mapについて図67を参照して説明する。ビデオストリーム用のEP\_mapは、stream\_PID、PTS\_EP\_start、および、RSPN\_EP\_startというデータを持つ。stream\_PIDは、ビデオストリームを伝送するトランスポートパケットのPIDを示す。PTS\_EP\_startは、ビデオストリームのシーケンスヘッダから始まるアクセスユニットのPTSを示す。RSPN\_EP\_startは、AVストリームの中でPTS\_EP\_startにより参照されるアクセスユニットの第1バイト目を含むソースパケットのアドレスを示す。

【0235】EP\_map\_for\_one\_stream\_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるビデオストリーム毎に作られる。Clipの中に複数のビデオストリームが存在する場合、EP\_mapは複数のEP\_map\_for\_one\_stream\_PID()を含んでも良い。

【0236】オーディオストリーム用のEP\_mapは、stream\_PID、PTS\_EP\_start、およびRSPN\_EP\_startというデータを持つ。stream\_PIDは、オーディオストリームを伝送するトランスポートパケットのPIDを示す。PTS\_EP\_startは、オーディオストリームのアクセスユニットのPTSを示す。RSPN\_EP\_startは、AVストリームの中でPTS\_EP\_startで参照されるアクセスユニットの第1バイト目を含むソースパケットのアドレスを示す。

【0237】EP\_map\_for\_one\_stream\_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるオーディオストリーム毎に作られる。Clipの中に複数のオーディオストリームが存在する場合、EP\_mapは複数のEP\_map\_for\_one\_stream\_PID()を含んでも良い。

【0238】EP\_mapとSTC\_Infoの関係を説明するに、1つのEP\_map\_for\_one\_stream\_PID()は、STCの不連続点に関係なく1つのテーブルに作られる。RSPN\_EP\_startの値とSTC\_Info()において定義されるRSPN\_STC\_startの値を比較する事により、それぞれのSTC\_sequenceに属するEP\_mapのデータの境界が分かる(図68を参照)。・EP\_mapは、同じPIDで伝送される連続したストリームの範囲に対して、1つのEP\_map\_for\_one\_stream\_PID()を持たねばならない。図69に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が

連続していないので、それぞれのプログラム毎にEP\_map\_for\_one\_stream\_PID()を持たねばならない。

【0239】図70は、EP\_mapのシンタクスを示す図である。図70に示したEP\_mapのシンタクスを説明するに、EP\_typeは、4ビットのフィールドであり、図71に示すように、EP\_mapのエントリーポイントタイプを示す。EP\_typeは、このフィールドに続くデータフィールドのセマンティクスを示す。Clipが1つ以上のビデオストリームを含む場合、EP\_typeは0('video')にセットされなければならない。または、Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、EP\_typeは1('audio')にセットされなければならない。

【0240】number\_of\_stream\_PIDsの16ビットのフィールドは、EP\_map()の中のnumber\_of\_stream\_PIDsを変数にもつfor-loopのループ回数を示す。stream\_PID(k)の16ビットのフィールドは、EP\_map\_for\_one\_stream\_PID(num\_EP\_entries(k))によって参照されるk番目のエレメンタリーストリーム(ビデオまたはオーディオストリーム)を伝送するトランスポートパケットのPIDを示す。EP\_typeが0('video')に等しい場合、そのエレメンタリーストリームはビデオストリームでなければならない。また、EP\_typeが1('audio')に等しい場合、そのエレメンタリーストリームはオーディオストリームでなければならない。

【0241】num\_EP\_entries(k)の16ビットのフィールドは、EP\_map\_for\_one\_stream\_PID(num\_EP\_entries(k))によって参照されるnum\_EP\_entries(k)を示す。EP\_map\_for\_one\_stream\_PID\_Start\_address(k): この32ビットのフィールドは、EP\_map()の中でEP\_map\_for\_one\_stream\_PID(num\_EP\_entries(k))が始まる相対バイト位置を示す。この値は、EP\_map()の第1バイト目からの大きさで示される。

【0242】padding\_wordは、EP\_map()のシンタクスにしたがって挿入されなければならない。XとYは、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値を取っても良い。

【0243】図72は、EP\_map\_for\_one\_stream\_PIDのシンタクスを示す図である。図72に示したEP\_map\_for\_one\_stream\_PIDのシンタクスを説明するに、PTS\_EP\_startの32ビットのフィールドのセマンティクスは、EP\_map()において定義されるEP\_typeにより異なる。EP\_typeが0('video')に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。EP\_typeが1('audio')に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。

【0244】RSPN\_EP\_startの32ビットのフィールドのセマンティクスは、EP\_map()において定義されるEP\_t

10

20

30

40

50

ypeにより異なる。EP\_typeが0 ('video')に等しい場合、このフィールドは、AVストリームの中でPTS\_EP\_startにより参照されるアクセスユニットのシーケンスヘッダの第1バイト目を含むソースパケットの相対アドレスを示す。または、EP\_typeが1 ('audio')に等しい場合、このフィールドは、AVストリームの中でPTS\_EP\_startにより参照されるアクセスユニットのオーディオフレームの第1バイト目を含むソースパケットの相対アドレスを示す。

【0245】RSPN\_EP\_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset\_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

$$SPN\_xxx = RSPN\_xxx - offset\_SPN$$

により算出される。シンタクスのfor-loopの中でRSPN\_EP\_startの値は、昇順に現れなければならない。

【0246】次に、TU\_mapについて、図73を参照して説明する。TU\_mapは、ソースパケットのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、1つの時間軸を作る。その時間軸は、TU\_map\_time\_axisと呼ばれる。TU\_map\_time\_axisの原点は、TU\_map()の中のoffset\_timeによって示される。TU\_map\_time\_axisは、offset\_timeから一定の単位に分割される。その単位を、time\_unitと称する。

【0247】AVストリームの中の各々のtime\_unitの中で、最初の完全な形のソースパケットのAVストリームファイル上のアドレスが、TU\_mapにストアされる。これらのアドレスを、RSPN\_time\_unit\_startと称する。TU\_map\_time\_axis上において、k (k>=0)番目のtime\_unitが始まる時刻は、TU\_start\_time(k)と呼ばれる。この値は次に基づいて算出される。

$TU\_start\_time(k) = offset\_time + k * time\_unit\_size$   
TU\_start\_time(k)は、45kHzの精度を持つ。

【0248】図74は、TU\_mapのシンタクスを示す図である。図74に示したTU\_mapのシンタクスを説明するに、offset\_timeの32bit長のフィールドは、TU\_map\_time\_axisに対するオフセットタイムを与える。この値は、Clipの中の最初のtime\_unitに対するオフセット時刻を示す。offset\_timeは、27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。AVストリームが新しいClipとして記録される場合、offset\_timeはゼロにセットされなければならない。

【0249】time\_unit\_sizeの32ビットフィールドは、time\_unitの大きさを与えるものであり、それは27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。time\_unit\_sizeは、1秒以下 (time\_unit\_size<=45000) にすることが良い。number\_of\_time\_unit\_entriesの32ビット

フィールドは、TU\_map()の中にストアされているtime\_unitのエントリ数を示す。

【0250】RSPN\_time\_unit\_startの32ビットフィールドは、AVストリームの中でそれぞれのtime\_unitが開始する場所の相対アドレスを示す。RSPN\_time\_unit\_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset\_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

$SPN\_xxx = RSPN\_xxx - offset\_SPN$

により算出される。シンタクスのfor-loopの中でRSPN\_time\_unit\_startの値は、昇順に現れなければならない。(k+1)番目のtime\_unitの中にソースパケットが何もない場合、(k+1)番目のRSPN\_time\_unit\_startは、k番目のRSPN\_time\_unit\_startと等しくなければならない。

【0251】図45に示したzzzzz.clipのシンタクス内のClipMarkについて説明する。ClipMarkは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置1）によってセットされるものであり、ユーザによってセットされるものではない。

【0252】図75は、ClipMarkのシンタクスを示す図である。図75に示したClipMarkのシンタクスを説明するに、version\_numberは、このClipMark()のバージョンナンバーを示す4個のキャラクター文字である。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0253】lengthは、このlengthフィールドの直後からClipMark()の最後までのClipMark()のバイト数を示す32ビットの符号なし整数である。number\_of\_clip\_marksは、ClipMarkの中にストアされているマークの個数を示す16ビットの符号なし整数。number\_of\_clip\_marksは、0であってもよい。mark\_typeは、マークのタイプを示す8ビットのフィールドであり、図76に示すテーブルに従って符号化される。

【0254】mark\_time\_stampは、32ビットフィールドであり、マークが指定されたポイントを示すタイムスタンプをストアする。mark\_time\_stampのセマンティクスは、図77に示すように、Playlist()の中のCPI\_typeにより異なる。

【0255】STC\_sequence\_idは、CPI()の中のCPI\_typeがEP\_map\_typeを示す場合、この8ビットのフィールドは、mark\_time\_stampが置かれているところのSTC連続区間のSTC\_sequence\_idを示す。CPI()の中のCPI\_typeがTU\_map\_typeを示す場合、この8ビットのフィールドは何も意味を持たず、ゼロにセットされる。character\_setの8ビットのフィールドは、mark\_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

【0256】name\_lengthの8ビットフィールドは、mark\_nameフィールドの中に示されるマーク名のバイト長を示す。mark\_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname\_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。mark\_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0257】ref\_thumbnail\_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref\_thumbnail\_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref\_thumbnail\_indexの値を用いて参照される。ref\_thumbnail\_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない。

【0258】図78は、図75に代わるClipMarkの他のシンタクスを示す図であり、図79は、その場合における、図76に代わるmark\_typeのテーブルの例を示す。reserved\_for\_maker\_IDは、mark\_typeが、0xC0から0xFFの値を示す時に、そのmark\_typeを定義しているメーカーのメーカーIDを示す16ビットのフィールドである。メーカーIDは、DVRフォーマットライセンサーが指定する。mark\_entry()は、マーク点に指定されたポイントを示す情報であり、そのシンタクスの詳細は後述する。representative\_picture\_entry()は、mark\_entry()によって示されるマークを代表する画像のポイントを示す情報であり、そのシンタクスの詳細は後述する。

【0259】ClipMarkは、ユーザがAVストリームを再生するときに、その内容を視覚的に検索できるようにするために用いられる。DVRプレーヤは、GUI(グラフィカルユーザインターフェース)を使用して、ClipMarkの情報をユーザに提示する。ClipMarkの情報を視覚的に表示するためには、mark\_entry()が示すピクチャよりもむしろrepresentative\_picture\_entry()が示すピクチャを示したほうが良い。

【0260】図80に、mark\_entry()とrepresentative\_picture\_entry()の例を示す。例えば、あるプログラムが開始してから、しばらくした後(数秒後)、そのプログラムの番組名(タイトル)が表示されるとする。ClipMarkを作るときは、mark\_entry()は、そのプログラムの開始ポイントに置き、representative\_picture\_entry()は、そのプログラムの番組名(タイトル)が表示されるポイントに置くようにしても良い。

【0261】DVRプレーヤは、representative\_picture\_entry()の画像をGUIに表示し、ユーザがその画像を指定すると、DVRプレーヤは、mark\_entry()の置かれたポイントから再生を開始する。

【0262】mark\_entry() および representative\_pic

ture\_entry()のシンタクスを、図81に示す。

【0263】mark\_time\_stampは、32ビットフィールドであり、mark\_entry()の場合はマークが指定されたポイントを示すタイムスタンプをストアし、またrepresentative\_picture\_entry()の場合、mark\_entry()によって示されるマークを代表する画像のポイントを示すタイムスタンプをストアする。

【0264】次に、ClipMarkを指定するために、PTSによるタイムスタンプベースの情報を使用するのではなく、アドレスベースの情報を使用する場合のmark\_entry()とrepresentative\_picture\_entry()のシンタクスの例を図82に示す。

【0265】RSPN\_ref\_EP\_startは、mark\_entry()の場合、AVストリームの中でマーク点のピクチャをデコードするためのストリームのエン트리ポイントを示すソースパケットの相対アドレスを示す。また、representative\_picture\_entry()の場合、mark\_entry()によって示されるマークを代表するピクチャをデコードするためのストリームのエン트리ポイントを示すソースパケットの相対アドレスを示す。RSPN\_ref\_EP\_startの値は、EP\_mapの中にRSPN\_EP\_startとしてストアされていなければならない。かつ、そのRSPN\_EP\_startに対応するPTS\_EP\_startの値は、EP\_mapの中で、マーク点のピクチャのPTSより過去で最も近い値でなければならない。

【0266】offset\_num\_picturesは、32ビットのフィールドであり、RSPN\_ref\_EP\_startにより参照されるピクチャから表示順序でマーク点で示されるピクチャまでのオフセットのピクチャ数を示す。この数は、ゼロからカウントされる。図83の例の場合、offset\_num\_picturesは6となる。

【0267】次に、ClipMarkを指定するために、アドレスベースの情報を使用する場合のmark\_entry()とrepresentative\_picture\_entry()のシンタクスの別の例を図84に示す。

【0268】RSPN\_mark\_pointは、mark\_entry()の場合、AVストリームの中で、そのマークが参照するアクセスユニットの第1バイト目を含むソースパケットの相対アドレスを示す。また、representative\_picture\_entry()の場合、mark\_entry()によって示されるマークを代表する符号化ピクチャの第1バイト目を含むソースパケットの相対アドレスを示す。

【0269】RSPN\_mark\_pointは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClip Information fileにおいて定義されるoffset\_SPNの値を初期値としてカウントされる。

【0270】図85を用いて、ClipMarkとEP\_mapの関係を説明する。この例の場合、EP\_mapが、エン트리ポイントのアドレスとしてI0、I1、I2を指定しており、これらのアドレスからシーケンスヘッダに続く1ピクチャが

開始しているとする。ClipMarkが、あるマークのアドレスとして、M1を指定している時、そのソースパケットから開始しているピクチャをデコードできるためには、M1のアドレスより前で最も近いエントリーポイントであるI1からデータを読み出し開始すれば良い。

【0271】MakersPrivateDataについては、図22を参照して既に説明したので、その説明は省略する。

【0272】次に、サムネイルインフォメーション (Thumbnail Information) について説明する。サムネイル画像は、menu.thmbファイルまたはmark.thmbファイルに10 ストアされる。これらのファイルは同じシンタクス構造であり、ただ1つのThumbnail()を持つ。menu.thmbファイルは、メニューサムネイル画像、すなわちVolumeを代表する画像、および、それぞれのPlaylistを代表する画像をストアする。すべてのメニューサムネイルは、ただ1つのmenu.thmbファイルにストアされる。

【0273】mark.thmbファイルは、マークサムネイル画像、すなわちマーク点を表すピクチャをストアする。すべてのPlaylistおよびClipに対するすべてのマークサムネイルは、ただ1つのmark.thmbファイルにストアさ10 れる。サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、Thumbnail()はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分の一つのtn\_blockに格納される。1つの画像データは連続したtn\_blockに格納される。tn\_blockの列には、使用されていないtn\_blockが存在してもよい。1つのサムネイル画像のバイト長は可変である。

【0274】図86は、menu.thmbとmark.thmbのシンタクスを示す図であり、図87は、図86に示したmenu.thmbとmark.thmbのシンタクス内のThumbnailのシンタクスを示す図である。図87に示したThumbnailのシンタクスについて説明するに、version\_numberは、このThumbnail()のバージョンナンバーを示す4個のキャラクター文字である。version\_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0275】lengthは、このlengthフィールドの直後からThumbnail()の最後までMakersPrivateData()のバイト数を示す32ビットの符号なし整数である。tn\_block\_start\_addressは、Thumbnail()の先頭のバイトからの10 相対バイト数を単位として、最初のtn\_blockの先頭バイトアドレスを示す32ビットの符号なし整数である。相対バイト数はゼロからカウントされる。number\_of\_thumbnailsは、Thumbnail()の中に含まれているサムネイル画像のエントリー数を与える16ビットの符号なし整数である。

【0276】tn\_block\_sizeは、1024バイトを単位として、1つのtn\_blockの大きさを与える16ビットの符号なし整数である。例えば、tn\_block\_size=1ならば、それは1つのtn\_blockの大きさが1024バイトであることを

示す。number\_of\_tn\_blocksは、このThumbnail()中のtn\_blockのエントリー数を表す116ビットの符号なし整数である。thumbnail\_indexは、このthumbnail\_indexフィールドから始まるforループ一回分のサムネイル情報で表されるサムネイル画像のインデクス番号を表す16ビットの符号なし整数である。thumbnail\_indexとして、0xFFFFという値を使用してはならない。thumbnail\_indexはUIAppInfoVolume()、UIAppInfoPlaylist()、PlaylistMark()、およびClipMark()の中のref\_thumbnail\_indexによって参照される。

【0277】thumbnail\_picture\_formatは、サムネイル画像のピクチャフォーマットを表す8ビットの符号なし整数で、図88に示すような値をとる。表中のDCFとPNGは"menu.thmb"内でのみ許される。マークサムネイルは、値"0x00" (MPEG-2 Video I-picture)をとらなければならない。

【0278】picture\_data\_sizeは、サムネイル画像のバイト長をバイト単位で示す32ビットの符号なし整数である。start\_tn\_block\_numberは、サムネイル画像のデータが始まるtn\_blockのtn\_block番号を表す16ビットの符号なし整数である。サムネイル画像データの先頭は、tb\_blockの先頭と一致していなければならない。tn\_block番号は、0から始まり、tn\_blockのfor-ループ中の変数kの値に関係する。

【0279】x\_picture\_lengthは、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す16ビットの符号なし整数である。y\_picture\_lengthは、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す16ビットの符号なし整数である。tn\_blockは、サムネイル画像がストアされる領域である。Thumbnail()の中のすべてのtn\_blockは、同じサイズ (固定長) であり、その大きさはtn\_block\_sizeによって定義される。

【0280】図89は、サムネイル画像データがどのようにtn\_blockに格納されるかを模式的に表した図である。図89のように、各サムネイル画像データはtn\_blockの先頭から始まり、1 tn\_blockを超える大きさの場合は、連続する次のtn\_blockを使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、10 削除といった編集に対して簡便な処理により対応する事ができるようになる。

【0281】次に、AVストリームファイルについて説明する。AVストリームファイルは、"M2TS"ディレクトリ (図14) にストアされる。AVストリームファイルには、2つのタイプがあり、それらは、Clip AVストリームとBridge-Clip AVストリームファイルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2トランスポートストリームファイルの構造でなければならない。

【0282】まず、DVR MPEG-2 トランスポートストリ



ームについて説明する。DVR MPEG-2トランスポートストリームの構造は、図90に示すようになっている。AVストリームファイルは、DVR MPEG2トランスポートストリームの構造を持つ。DVR MPEG2トランスポートストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、6144 バイト (2048<sup>3</sup> バイト)である。Aligned unitは、ソースパケットの第1バイト目から始まる。ソースパケットは、192バイト長である。一つのソースパケットは、TP\_extra\_headerとトランスポートパケットから成る。TP\_extra\_headerは、4バイト長であり、またトランスポートパケットは、188バイト長である。

【0283】1つのAligned unitは、32個のソースパケットから成る。DVR MPEG2トランスポートストリームの中の最後のAligned unitも、また32個のソースパケットから成る。よって、DVR MPEG2トランスポートストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トランスポートストリームのトランスポートパケットの数が32の倍数でない時、ヌルパケット (PID=0x1FFFのトランスポートパケット) を持ったソースパケットを最後のAligned unitに使用しなければならない。ファイルシステムは、DVR MPEG2トランスポートストリームに余分な情報を付加してはならない。

【0284】図91に、DVR MPEG-2トランスポートストリームのレコーダモデルを示す。図91に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

【0285】MPEG-2トランスポートストリームの入力タイミングについて説明する。入力MPEG2トランスポートストリームは、フルトランスポートストリームまたはパシャルトランスポートストリームである。入力されるMPEG2トランスポートストリームは、ISO/IEC13818-1またはISO/IEC13818-9に従っていなければならない。MPEG2トランスポートストリームのi番目のバイトは、T-STD (ISO/IEC 13818-1で規定されるTransport stream system target decoder) 51とソースパケットタイザ (source packetizer) 54へ、時刻t(i)に同時に入力される。Rp<sub>k</sub>は、トランスポートパケットの入力レートの瞬時的な最大値である。

【0286】27MHz PLL 52は、27MHzクロックの周波数を発生する。27MHzクロックの周波数は、MPEG-2トランスポートストリームのPCR (Program Clock Reference)の値にロックされる。アライバルタイムクロックカウンタ (arrival time clock counter) 53は、27MHzの周波数のパルスをカウントするバイナリーカウンタである。Arrival\_time\_clock(i)は、時刻t(i)におけるarrival time clockcounter 53のカウント値である。

【0287】source packetizer 54は、すべてのトラ

ンスポートパケットにTP\_extra\_headerを付加し、ソースパケットを作る。Arrival\_time\_stampは、トランスポートパケットの第1バイト目がT-STD 51とソースパケットタイザ 54の両方へ到着する時刻を表す。Arrival\_time\_stamp(k)は、次式で示されるようにArrival\_time\_clock(k)のサンプル値であり、ここで、kはトランスポートパケットの第1バイト目を示す。

$$\text{arrival\_time\_stamp}(k) = \text{arrival\_time\_clock}(k) \cdot 2^{30}$$

【0288】2つの連続して入力されるトランスポートパケットの時間間隔が、 $2^{30}/27000000$ 秒 (約40秒) 以上になる場合、その2つのトランスポートパケットのarrival\_time\_stampの差分は、 $2^{30}/27000000$ 秒になるようにセットされるべきである。レコーダは、そのような場合に備えてある。

【0289】スムージングバッファ (smoothing buffer) 55は、入力トランスポートストリームのビットレートをスムージングする。スムージングバッファ 55は、オーバーフローしてはならない。R<sub>max</sub>は、スムージングバッファ 55が空でない時のスムージングバッファ 55からのソースパケットの出力ビットレートである。スムージングバッファ 55が空である時、スムージングバッファ 55からの出力ビットレートはゼロである。

【0290】次に、DVR MPEG-2トランスポートストリームのレコーダモデルのパラメータについて説明する。R<sub>max</sub>という値は、AVストリームファイルに対応するClipInfo()において定義されるTS\_recording\_rateによって与えられる。この値は、次式により算出される。

$$R_{\text{max}} = \text{TS\_recording\_rate} \cdot 192/188$$

TS\_recording\_rateの値は、bytes/secondを単位とする大きさである。

【0291】入力トランスポートストリームがSESFトランスポートストリームの場合、Rp<sub>k</sub>は、AVストリームファイルに対応するClipInfo()において定義されるTS\_recording\_rateに等しくなければならない。入力トランスポートストリームがSESFトランスポートストリームでない場合、この値はMPEG-2 transport streamのデスクリプター、例えばmaximum\_bitrate\_descriptorやpartial\_transport\_stream\_descriptorなど、において定義される値を参照しても良い。

【0292】入力トランスポートストリームがSESFトランスポートストリームの場合、スムージングバッファ 55の大きさ (smoothing buffer size) はゼロである。入力トランスポートストリームがSESFトランスポートストリームでない場合、スムージングバッファ 55の大きさはMPEG-2 transport streamのデスクリプター、例えばsmoothing\_buffer\_descriptor、short\_smoothing\_buffer\_descriptor、partial\_transport\_stream\_descriptorなどにおいて定義される値を参照しても良い。

【0293】記録機 (レコーダ) および再生機 (プレーヤ) は、十分なサイズのバッファを用意しなければならない

ない。デフォルトのバッファサイズは、1536 bytes である。

【0294】次に、DVR MPEG-2トランスポートストリームのプレーヤモデルについて説明する。図92は、DVR MPEG-2トランスポートストリームのプレーヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

【0295】27MHz X-tal (クリスタル発振器) 61 は、27MHzの周波数を発生する。27MHz周波数の誤差範囲は、 $\pm 30$  ppm (27000000  $\pm$  810 Hz)でなければならない。arrival time clock counter 62は、27MHzの周波数のパルスをカウントするバイナリカウンタである。arrival\_time\_clock(i)は、時刻t(i)におけるarrival time clock counter 62のカウント値である。

【0296】smoothing buffer 64において、Rmaxは、スムージングバッファ64がフルでない時のスムージングバッファ64へのソースパケットの入力ビットレートである。スムージングバッファ64がフルである時、スムージングバッファ64への入力ビットレートはゼロである。

【0297】MPEG-2トランスポートストリームの出力タイミングを説明するに、現在のソースパケットのarrival\_time\_stampがarrival\_time\_clock(i)のLSB 30ビットの値と等しい時、そのソースパケットのトランスポートパケットは、スムージングバッファ64から引き抜かれる。Rpkは、トランスポートパケットレートの瞬時的な最大値である。スムージングバッファ64は、アンダーフローしてはならない。

【0298】DVR MPEG-2トランスポートストリームのプレーヤモデルのパラメータについては、上述したDVR MPEG-2トランスポートストリームのレコーダモデルのパラメータと同一である。

【0299】図93は、Source packetのシンタクスを示す図である。transport\_packet()は、ISO/IEC 13818-1で規定されるMPEG-2トランスポートパケットである。図93に示したSource packetのシンタクス内のTP\_Extra\_headerのシンタクスを図94に示す。図94に示したTP\_Extra\_headerのシンタクスについて説明するに、copy\_permission\_indicatorは、トランスポートパケットのペイロードのコピー制限を表す整数である。コピー制限は、copy free, no more copy, copy once, またはcopy prohibitedとすることができる。図95は、copy\_permission\_indicatorの値と、それらによって指定されるモードの関係を示す。

【0300】copy\_permission\_indicatorは、すべてのトランスポートパケットに付加される。IEEE1394デジタルインターフェースを使用して入力トランスポートストリームを記録する場合、copy\_permission\_indicatorの

値は、IEEE1394 isochronous packet headerの中のEMI (Encryption Mode Indicator)の値に関連付けても良い。IEEE1394デジタルインターフェースを使用しないで入力トランスポートストリームを記録する場合、copy\_permission\_indicatorの値は、トランスポートパケットの中に埋め込まれたCCIの値に関連付けても良い。アナログ信号入力をセルフエンコードする場合、copy\_permission\_indicatorの値は、アナログ信号のCGMS-Aの値に関連付けても良い。

【0301】arrival\_time\_stampは、次式 
$$\text{arrival\_time\_stamp}(k) = \text{arrival\_time\_clock}(k) \% 2^{30}$$
 において、arrival\_time\_stampによって指定される値を持つ整数値である。

【0302】Clip AVストリームの定義をするに、Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。arrival\_time\_clock(i)は、Clip AVストリームの中で連続して増加しなければならない。Clip AVストリームの中にシステムタイムベース (STCベース) の不連続点が存在したとしても、そのClip AVストリームのarrival\_time\_clock(i)は、連続して増加しなければならない。

【0303】Clip AVストリームの中の開始と終了の間のarrival\_time\_clock(i)の差分の最大値は、26時間でなければならない。この制限は、MPEG2トランスポートストリームの中にシステムタイムベース (STCベース) の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS(Presentation Time Stamp)が決して現れないことを保証する。MPEG2システムズ規格は、PTSのラップアラウンド周期を233/90000秒(約26.5時間)と規定している。

【0304】Bridge-Clip AVストリームの定義をするに、Bridge-Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。Bridge-Clip AVストリームは、1つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランスポートストリームは、後述する符号化の制限に従わなければならない、かつ後述するDVR-STDに従わなければならない。

【0305】本実施の形態においては、編集におけるPlayItem間のビデオとオーディオのシームレス接続をサポートする。PlayItem間をシームレス接続にすることは、プレーヤ/レコーダに"データの連続供給"と"シームレスな復号処理"を保証する。"データの連続供給"とは、ファイルシステムが、デコーダにバッファのアンダーフローを起こさせる事のないように必要なビットレートでデータを供給する事を保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるように、データが十分な大きさの連続

したブロック単位でストアされるようにする。

【0306】"シームレスな復号処理"とは、プレーヤが、デコーダの再生出力にポーズやギャップを起こさせる事なく、ディスクに記録されたオーディオビデオデータを表示できることである。

【0307】シームレス接続されているPlayItemが参照するAVストリームについて説明する。先行するPlayItemと現在のPlayItemの接続が、シームレス表示できるように保証されているかどうかは、現在のPlayItemにおいて定義されているconnection\_conditionフィールドから判断することができる。PlayItem間のシームレス接続は、Bridge-Clipを使用する方法と使用しない方法がある。

【0308】図96は、Bridge-Clipを使用する場合の先行するPlayItemと現在のPlayItemの関係を示している。図96においては、プレーヤが読み出すストリームデータが、影をつけて示されている。図96に示したTS1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN\_arrival\_time\_discontinuityより前の影を付けられたストリームデータから成る。

【0309】TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN\_time (図96においてIN\_time1で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから、RSPN\_exit\_from\_previous\_clipで参照されるソースパケットまでのストリームデータである。TS1に含まれるBridge-ClipのRSPN\_arrival\_time\_discontinuityより前の影を付けられたストリームデータは、Bridge-Clipの最初のソースパケットから、RSPN\_arrival\_time\_discontinuityで参照されるソースパケットの直前のソースパケットまでのストリームデータである。

【0310】また、図96におけるTS2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN\_arrival\_time\_discontinuity以後の影を付けられたストリームデータから成る。TS2に含まれるBridge-ClipのRSPN\_arrival\_time\_discontinuity以後の影を付けられたストリームデータは、RSPN\_arrival\_time\_discontinuityで参照されるソースパケットから、Bridge-Clipの最後のソースパケットまでのストリームデータである。TS2のClip2の影を付けられたストリームデータは、RSPN\_enter\_to\_current\_clipで参照されるソースパケットから、現在のPlayItemのOUT\_time (図96においてOUT\_time2で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【0311】図97は、Bridge-Clipを使用しない場合の先行するPlayItemと現在のPlayItemの関係を示している。この場合、プレーヤが読み出すストリームデータは、影をつけて示されている。図97におけるTS1は、Clip1 (Clip AVストリーム) の影を付けられたストリーム

データから成る。TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN\_time (図97においてIN\_time1で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから始まり、Clip1の最後のソースパケットまでのデータである。また、図97におけるTS2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータから成る。

【0312】TS2のClip2の影を付けられたストリームデータは、Clip2の最初のソースパケットから始まり、現在のPlayItemのOUT\_time (図97においてOUT\_time2で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【0313】図96と図97において、TS1とTS2は、ソースパケットの連続したストリームである。次に、TS1とTS2のストリーム規定と、それらの間の接続条件について考える。まず、シームレス接続のための符号化制限について考える。トランスポートストリームの符号化構造の制限として、まず、TS1とTS2の中に含まれるプログラムの数は、1でなければならない。TS1とTS2の中に含まれるビデオストリームの数は、1でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、2以下でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、等しくなければならない。TS1および/またはTS2の中に、上記以外のエレメンタリーストリームまたはプライベートストリームが含まれていても良い。

【0314】ビデオビットストリームの制限について説明する。図98は、ピクチャの表示順序で示すシームレス接続の例を示す図である。接続点においてビデオストリームをシームレスに表示できるためには、OUT\_time1 (Clip1のOUT\_time) の後とIN\_time2 (Clip2のIN\_time) の前に表示される不必要なピクチャは、接続点付近のClipの部分的なストリームを再エンコードするプロセスにより、除去されなければならない。

【0315】図98に示したような場合において、BridgeSequenceを使用してシームレス接続を実現する例を、図99に示す。RSPN\_arrival\_time\_discontinuityより前のBridge-Clipのビデオストリームは、図98のClip1のOUT\_time1に対応するピクチャまでの符号化ビデオストリームから成る。そして、そのビデオストリームは先行するClip1のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

【0316】同様にして、RSPN\_arrival\_time\_discontinuity以後のBridge-Clipのビデオストリームは、図98のClip2のIN\_time2に対応するピクチャ以後の符号化ビデオストリームから成る。そして、そのビデオストリームは、正しくデコード開始する事ができて、これに続く

Clip2のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードしなければならない、それ以外のピクチャはオリジナルのClipからコピーすることができる。

【0317】図98に示した例の場合にBridgeSequenceを使用しないでシームレス接続を実現する例を図100に示す。Clip1のビデオストリームは、図98のOUT\_time1に対応するピクチャまでの符号化ビデオストリームから成り、それは、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。同様に、Clip2のビデオストリームは、図98のClip2のIN\_time2に対応するピクチャ以後の符号化ビデオストリームから成り、それは、一つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

【0318】ビデオストリームの符号化制限について説明するに、まず、TS1とTS2のビデオストリームのフレームレートは、等しくなければならない。TS1のビデオストリームは、sequence\_end\_codeで終端しなければならない。TS2のビデオストリームは、Sequence Header、GOP Header、そしてI-ピクチャで開始しなければならない。TS2のビデオストリームは、クローズドGOPで開始しなければならない。

【0319】ビットストリームの中で定義されるビデオプレゼンテーションユニット（フレームまたはフィールド）は、接続点を挟んで連続でなければならない。接続点において、フレームまたはフィールドのギャップがあってはならない。接続点において、トップ?ボトム?のフィールドシーケンスは連続でなければならない。3-2プルダウンを使用するエンコードの場合は、"top\_field\_first" および "repeat\_first\_field" フラグを書き換える必要があるかもしれない、またはフィールドギャップの発生を防ぐために局所的に再エンコードするようにしても良い。

【0320】オーディオビットストリームの符号化制限について説明するに、TS1とTS2のオーディオのサンプリング周波数は、同じでなければならない。TS1とTS2のオーディオの符号化方法（例、MPEG1レイヤ2、AC-3、SESF LPCM、AAC）は、同じでなければならない。

【0321】次に、MPEG-2トランスポートストリームの符号化制限について説明するに、TS1のオーディオストリームの最後のオーディオフレームは、TS1の最後の表示ピクチャの表示終了時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS2のオーディオストリームの最初のオーディオフレームは、TS2の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

【0322】接続点において、オーディオプレゼンター

ションユニットのシーケンスにギャップがあってはならない。図101に示すように、2オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあっても良い。TS2のエレメンタリーストリームを伝送する最初のパケットは、ビデオパケットでなければならない。接続点におけるトランスポートストリームは、後述するDVR-STDに従わなくてはならない。

【0323】ClipおよびBridge-Clipの制限について説明するに、TS1とTS2は、それぞれの中にアライバルタイムベースの不連続点を含んではならない。

【0324】以下の制限は、Bridge-Clipを使用する場合にのみ適用される。TS1の最後のソースパケットとTS2の最初のソースパケットの接続点においてのみ、Bridge-Clip AVストリームは、ただ1つのアライバルタイムベースの不連続点を持つ。ClipInfo()において定義されるRSPN\_arrival\_time\_discontinuityが、その不連続点のアドレスを示し、それはTS2の最初のソースパケットを参照するアドレスを示さなければならない。

【0325】BridgeSequenceInfo()において定義されるRSPN\_exit\_from\_previous\_Clipによって参照されるソースパケットは、Clip1の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。BridgeSequenceInfo()において定義されるRSPN\_enter\_to\_current\_Clipによって参照されるソースパケットは、Clip2の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。

【0326】PlayItemの制限について説明するに、先行するPlayItemのOUT\_time（図96、図97において示されるOUT\_time1）は、TS1の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在のPlayItemのIN\_time（図96、図97において示されるIN\_time2）は、TS2の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

【0327】Bridge-Clipを使用する場合のデータアロケーションの制限について、図102を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1（Clip AVストリームファイル）とClip2（Clip AVストリームファイル）に接続されるBridge-Clip AVストリームを、データアロケーション規定を満たすように配置することによって行われなければならない。

【0328】RSPN\_exit\_from\_previous\_Clip以前のClip1（Clip AVストリームファイル）のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN\_exit\_from\_previous\_Clipが選択されなければならない。Bridge-Clip AVストリームのデータ長は、ハーフフラグメント以上の連続領域に配置されるよう

に、選択されなければならない。RSPN\_enter\_to\_current\_Clip以後のClip2 (Clip AVストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN\_enter\_to\_current\_Clipが選択されなければならない。

【0329】Bridge-Clipを使用しないでシームレス接続する場合のデータアロケーションの制限について、図103を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip AVストリームファイル) の最後の部分とClip2 (Clip AVストリームファイル) の最初の部分を、データアロケーション規定を満たすように配置することによって行われなければならない。

【0330】Clip1 (Clip AVストリームファイル) の最後のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。Clip2 (Clip AVストリームファイル) の最初のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。

【0331】次に、DVR-STDについて説明する。DVR-STDは、DVR MPEG2トランスポートストリームの生成および検証の際におけるデコード処理をモデル化するための概念モデルである。また、DVR-STDは、上述したシームレス接続された2つのPlayItemによって参照されるAVストリームの生成および検証の際におけるデコード処理をモデル化するための概念モデルでもある。

【0332】DVR-STDモデルを図104に示す。図104に示したモデルには、DVR MPEG-2トランスポートストリームプレーヤモデルが構成要素として含まれている。n, TBn, MBn, EBn, TBSys, Bsys, Rxn, Rbxn, Rxsys, Dn, Dsys, OnおよびPn(k)の表記方法は、ISO/IEC13818-1のT-STDに定義されているものと同じである。すなわち、次の通りである。nは、エレメンタリーストリームのインデクス番号である。TBnは、エレメンタリーストリームnのトランスポートバッファである。

【0333】MBnは、エレメンタリーストリームnの多重バッファである。ビデオストリームについてのみ存在する。EBnは、エレメンタリーストリームnのエレメンタリーストリームバッファである。ビデオストリームについてのみ存在する。TBSysは、復号中のプログラムのシステム情報のための入力バッファである。Bsysは、復号中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。Rxnは、データがTBnから取り除かれる伝送レートである。Rbxnは、PESパケットペイロードがMBnから取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

【0334】Rxsysは、データがTBSysから取り除かれる伝送レートである。Dnは、エレメンタリーストリームnのデコーダである。Dsysは、復号中のプログラムのシ

テム情報に関するデコーダである。Onは、ビデオストリームnのre-ordering bufferである。Pn(k)は、エレメンタリーストリームnのk番目のプレゼンテーションユニットである。

【0335】DVR-STDのデコーディングプロセスについて説明する。単一のDVR MPEG-2トランスポートストリームを再生している間は、トランスポートパケットをTB1, TBnまたはTBSysのバッファへ入力するタイミングは、ソースパケットのarrival\_time\_stampにより決定される。TB1, MB1, EB1, TBn, Bn, TBSysおよびBsysのバッファリング動作の規定は、ISO/IEC 13818-1に規定されているT-STDと同じである。復号動作と表示動作の規定もまた、ISO/IEC 13818-1に規定されているT-STDと同じである。

【0336】シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される2つのAVストリームの再生について説明をすることにし、以後の説明では、上述した(例えば、図96に示した)TS1とTS2の再生について説明する。TS1は、先行するストリームであり、TS2は、現在のストリームである。

【0337】図105は、あるAVストリーム (TS1) からそれにシームレスに接続された次のAVストリーム (TS2) へと移る時のトランスポートパケットの入力、復号、表示のタイミングチャートを示す。所定のAVストリーム (TS1) からそれにシームレスに接続された次のAVストリーム (TS2) へと移る間には、TS2のアライバルタイムベースの時間軸 (図105においてATC2で示される) は、TS1のアライバルタイムベースの時間軸 (図105においてATC1で示される) と同じでない。

【0338】また、TS2のシステムタイムベースの時間軸 (図105においてSTC2で示される) は、TS1のシステムタイムベースの時間軸 (図105においてSTC1で示される) と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあっても良い。

【0339】DVR-STD への入力タイミングについて説明する。時刻T<sub>1</sub>までの時間、すなわち、TS1の最後のビデオパケットがDVR-STDのTB1に入力終了するまでは、DVR-STDのTB1, TBn またはTBSysのバッファへの入力タイミングは、TS1のソースパケットのarrival\_time\_stampによって決定される。

【0340】TS1の残りのパケットは、TS\_recording\_rate(TS1)のビットレートでDVR-STDのTBnまたはTBSysのバッファへ入力されなければならない。ここで、TS\_recording\_rate(TS1)は、Clip1に対応するClipInfo()において定義されるTS\_recording\_rateの値である。TS1の最後のバイトがバッファへ入力する時刻は、時刻T<sub>2</sub>であ

る。従って、時刻 $T_1$ から $T_2$ までの区間では、ソースパケットのarrival\_time\_stampは無視される。

【0341】 $N1$ を $TS1$ の最後のビデオパケットに続く $TS1$ のトランスポートパケットのバイト数とすると、時刻 $T_1$ 乃至 $T_2$ までの時間 $DT1$ は、 $N1$ バイトが $TS\_recording\_rate(TS1)$ のビットレートで入力終了するために必要な時間であり、次式により算出される。

$$DT1 = T_2 - T_1 = N1 / TS\_recording\_rate(TS1)$$

時刻 $T_1$ 乃至 $T_2$ までの間は、 $RXn$ と $RXsys$ の値は共に、 $TS\_recording\_rate(TS1)$ の値に変化する。このルール以外のバッファリング動作は、 $T$ -STDと同じである。

【0342】 $T_2$ の時刻において、arrival time clock counterは、 $TS2$ の最初のソースパケットのarrival\_time\_stampの値にリセットされる。 $DVR$ -STDの $TB1$ 、 $TBn$ または $TBsys$ のバッファへの入力タイミングは、 $TS2$ のソースパケットのarrival\_time\_stampによって決定される。 $RXn$ と $RXsys$ は共に、 $T$ -STDにおいて定義されている値に変化する。

【0343】付加的なオーディオバッファリングおよびシステムデータバッファリングについて説明するに、オーディオデコードとシステムデコードは、時刻 $T_1$ から $T_2$ までの区間の入力データを処理することができるように、 $T$ -STDで定義されるバッファ量に加えて付加的なバッファ量(約1秒分のデータ量)が必要である。

【0344】ビデオのプレゼンテーションタイミングについて説明するに、ビデオプレゼンテーションユニットの表示は、接続点を通して、ギャップなしに連続でなければならない。ここで、 $STC1$ は、 $TS1$ のシステムタイムベースの時間軸(図105では $STC1$ と図示されている)とし、 $STC2$ は、 $TS2$ のシステムタイムベースの時間軸(図97では $STC2$ と図示されている。正確には、 $STC2$ は、 $TS2$ の最初のPCRが $T$ -STDに入力した時刻から開始する。)とする。

【0345】 $STC1$ と $STC2$ の間のオフセットは、次のように決定される。 $PTS_{end}^1$ は、 $TS1$ の最後のビデオプレゼンテーションユニットに対応する $STC1$ 上のPTSであり、 $PTS_{start}^2$ は、 $TS2$ の最初のビデオプレゼンテーションユニットに対応する $STC2$ 上のPTSであり、 $T_{pp}$ は、 $TS1$ の最後のビデオプレゼンテーションユニットの表示期間とすると、2つのシステムタイムベースの間のオフセット $STC\_delta$ は、次式により算出される。

$$STC\_delta = PTS_{end}^1 + T_{pp} - PTS_{start}^2$$

【0346】オーディオのプレゼンテーションのタイミングについて説明するに、接続点において、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあっても良く、それは0乃至2オーディオフレーム未満である(図105に図示されている"audio overlap"を参照)。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに

再同期することは、プレーヤ側により設定されることである。

【0347】 $DVR$ -STDのシステムタイムクロックについて説明するに、時刻 $T_2$ において、 $TS1$ の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻 $T_2$ から $T_3$ の間にオーバーラップしていても良い。この区間では、 $DVR$ -STDは、システムタイムクロックを古いタイムベースの値( $STC1$ )と新しいタイムベースの値( $STC2$ )の間で切り替える。 $STC2$ の値は、次式により算出される。

$$STC2 = STC1 - STC\_delta$$

【0348】バッファリングの連続性について説明する。 $STC1_{video\_end}^1$ は、 $TS1$ の最後のビデオパケットの最後のバイトが $DVR$ -STDの $TB1$ へ到着する時のシステムタイムベース $STC1$ 上の $STC$ の値である。 $STC2_{video\_start}^2$ は、 $TS2$ の最初のビデオパケットの最初のバイトが $DVR$ -STDの $TB1$ へ到着する時のシステムタイムベース $STC2$ 上の $STC$ の値である。 $STC2_{video\_end}^1$ は、 $STC1_{video\_end}^1$ の値をシステムタイムベース $STC2$ 上の値に換算した値である。 $STC2_{video\_end}^1$ は、次式により算出される。

$$STC2_{video\_end}^1 = STC1_{video\_end}^1 - STC\_delta$$

【0349】 $DVR$ -STDに従うために、次の2つの条件を満たす事が要求される。まず、 $TS2$ の最初のビデオパケットの $TB1$ への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

$$STC2_{video\_start}^2 > STC2_{video\_end}^1 + \Delta T_1$$

この不等式が満たされるように、Clip1および、または、Clip2の部分的なストリームを再エンコードおよび、または、再多重化する必要がある場合は、その必要に応じて行われる。

【0350】次に、 $STC1$ と $STC2$ を同じ時間軸上に換算したシステムタイムベースの時間軸上において、 $TS1$ からのビデオパケットの入力とそれに続く $TS2$ からのビデオパケットの入力は、ビデオバッファをオーバーフローおよびアンダーフローさせてはならない。

【0351】このようなシンタクス、データ構造、規則に基づく事により、記録媒体に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

【0352】なお、本実施の形態は、多重化ストリームとしてMPEG2トランスポートストリームを例にして説明しているが、これに限らず、MPEG2プログラムストリームや米国のDirecTVサービス(商標)で使用されているDSSトランスポートストリームについても適用することが可能である。

【0353】次に、mark\_entry()およびrepresentative\_picture\_entry()のシンタクスが、図81に示される

ような構成である場合における、マーク点で示されるシーンの頭出し再生を行う場合の処理について、図106のフローチャートを参照して、説明する。

【0354】最初にステップS1において、記録再生装置1の制御部23は、記録媒体100から、DVRトランスポートストリームファイルのデータデースであるEP\_Map(図70)、STC\_Info(図52)、Program\_Info(図54)、およびClipMark(図78)を読み出す。

【0355】ステップS2において、制御部23は、ClipMark(図78)のrepresentative\_picture\_entry(図81)、またはref\_thumbnail\_indexで参照されるピクチャからサムネイルのリストを作成し、ユーザインターフェース入出力としての端子24から出力し、GUIのメニュー画面上に表示させる。この場合、ref\_thumbnail\_indexが有効な値を持つ場合、representative\_picture\_entryよりref\_thumbnail\_indexが優先される。

【0356】ステップS3において、ユーザが再生開始点のマーク点を指定する。これは、例えば、GUIとして表示されたメニュー画面上の中からユーザがサムネイル画像を選択することで行われる。制御部23は、この選択操作に対応して、指定されたサムネイルに対応づけられているマーク点を取得する。

【0357】ステップS4において、制御部23は、ステップS3で指定されたmark\_entry(図81)のmark\_time\_stampのPTSと、STC\_sequence\_idを取得する。

【0358】ステップS5において、制御部23は、STC\_Info(図52)から、ステップS4で取得したSTC\_sequence\_idに対応するSTC時間軸が開始するソースパケット番号を取得する。

【0359】ステップS6において、制御部23は、ステップS5で取得したSTC時間軸が開始するパケット番号と、ステップS4で取得したマーク点のPTSから、マーク点のPTSより時間的に前で、かつ、最も近いエントリポイント(1ピクチャ)のあるソースパケット番号を取得する。

【0360】ステップS7において、制御部23は、ステップS6で取得したエントリポイントのあるソースパケット番号から、トランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0361】ステップS8において、制御部23は、AVデコーダ27を制御し、ステップS4で取得したマーク点のPTSのピクチャから表示を開始させる。

【0362】以上の動作を、図107乃至109を参照してさらに説明する。

【0363】いま、図107に示されているように、DVRトランスポートストリームファイルは、STC\_sequence\_id=id0のSTC時間軸を有し、その時間軸が開始するソースパケット番号は、シーン開始点Aのソースパケット番号より小さいものとする。そして、ソースパケット番号BからCまでの間に、CM(コマーシャル)が挿入されて

いるものとする。

【0364】このとき、図70に示されるEP\_Mapに対応するEP\_Mapには、図108に示されるように、RSPN\_EP\_startで示されるA、B、Cに対応して、それぞれのPTSが、PTS\_EP\_startとして、PTS(A)、PTS(B)、PTS(C)として登録される。

【0365】また、図109に示されるように、図78のClipMarkに対応するClipMarkには、図109に示されるように、シーンスタート、CMスタート、およびCMエンドを表すマークタイプ(図79)0x92、0x94、0x95の値に対応して、mark\_entryとrepresentative\_picture\_entryが記録される。

【0366】mark\_entryのMark\_Time\_stampとしては、シーンスタート、CMスタート、およびCMエンドに対応して、それぞれPTS(a1)、PTS(b0)、PTS(c0)が登録されており、それぞれのSTC\_sequence\_idは、いずれもid0とされている。

【0367】同様に、Representative\_picture\_entryのMark\_Time\_stampとして、シーンスタート、CMスタート、およびCMエンドに対応して、それぞれPTS(a2)、PTS(b0)、PTS(c0)が登録されており、それらはいずれもSTC\_sequence\_idが、id0とされている。

【0368】PTS(A)<PTS(a1)の場合、ステップS6において、パケット番号Aが取得され、ステップS7において、パケット番号Aから始まるトランスポートストリームが、AVデコーダ27に供給され、ステップS8において、PTS(a1)のピクチャから表示が開始される。

【0369】次に、図110のフローチャートを参照して、mark\_entryとrepresentative\_picture\_entryのシンタクスが、図81に示されるような構成である場合におけるCMスキップ再生の処理について、図110のフローチャートを参照して説明する。

【0370】ステップS21において、制御部23は、EP\_map(図70)、STC\_Info(図52)、Program\_Info(図54)、およびClipMark(図78)を記録媒体100から読み出す。ステップS22において、ユーザは、ユーザインターフェース入出力としての端子24からCMスキップ再生を指定する。

【0371】ステップS23において、制御部23は、マークタイプ(図79)がCM開始点(0x94)であるマーク情報のPTSと、CM終了点(0x95)であるマーク情報のPTS、並びに対応するSTC\_sequence\_idを取得する(図81)。

【0372】ステップS24において、制御部23は、STC\_Info(図52)からCM開始点と終了点の、STC\_sequence\_idに対応するSTC時間軸が開始するソースパケット番号を取得する。

【0373】ステップS25において、制御部23は、記録媒体100からトランスポートストリームを読み出させ、それをAVデコーダ27に供給し、デコードを開始

させる。

【0374】ステップS26において、制御部23は、現在の表示画像がCM開始点のPTSの画像か否かを調べる。現在の表示画像がCM開始点のPTSの画像でない場合には、ステップS27に進み、制御部23は、画像の表示が継続される。その後、処理はステップS25に戻り、それ以降の処理が繰り返し実行される。

【0375】ステップS26において、現在の表示画像がCM開始点のPTSの画像であると判定された場合、ステップS28に進み、制御部23は、AVデコーダ27を制御し、デコードおよび表示を停止させる。

【0376】次に、ステップS29において、制御部23は、CM終了点のSTC\_sequence\_idに対応するSTC時間軸が開始するパケット番号を取得し、そのパケット番号と、ステップS23の処理で取得したCM終了点のPTSとから、その点のPTSより時間的に前で、かつ、最も近いエントリーポイントのあるソースパケット番号を取得する。

【0377】ステップS30において、制御部23は、ステップS29の処理で取得したエントリーポイントのあるソースパケット番号から、トランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0378】ステップS31において、制御部23は、AVデコーダ27を制御し、CM終了点のPTSのピクチャから表示を再開させる。

【0379】図107乃至図109を参照して、以上の動作をさらに説明すると、CM開始点とCM終了点は、この例の場合、STC\_sequence\_id=id0という共通のSTC時間軸上に存在し、そのSTC時間軸が開始するソースパケット番号は、シーンの開始点のソースパケット番号Aより小さいものとされている。

【0380】トランスポートストリームがデコードされ、ステップS26で、表示時刻がPTS(b0)になったと判定された場合（CM開始点であると判定された場合）、AVデコーダ27により表示が停止される。そして、PTS(c)<PTS(c0)の場合、ステップS30でパケット番号Cのデータから始まるストリームからデコードが再開され、ステップS31において、PTS(c0)のピクチャから表示が再開される。

【0381】なお、この方法は、CMスキップ再生に限らず、一般的にClipMarkで指定される2点間のシーンをスキップして再生する場合にも、適用可能である。

【0382】次に、mark\_entryとrepresentative\_picture\_entryが、図82に示すシンタクス構造である場合における、マーク点で示されるCMの頭出し再生処理について、図112のフローチャートを参照して説明する。

【0383】ステップS41において、制御部23は、EP\_map（図70）、STC\_Info（図52）、Program\_Info（図54）、およびClipMark（図78）の情報を取得する。

【0384】次にステップS42において、制御部23は、ステップS41で読み出したClipMark（図78）に含まれるrepresentative\_picture\_entry（図82）またはref\_thumbnail\_indexで参照されるピクチャからサムネイルのリストを生成し、GUIのメニュー画面上に表示させる。ref\_thumbnail\_indexが有効な値を有する場合、representative\_picture\_entryよりref\_thumbnail\_indexが優先される。

【0385】ステップS43において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、ステップS42の処理で表示されたメニュー画面上の中から、ユーザがサムネイル画像を選択し、そのサムネイルに対応づけられているマーク点を指定することで行われる。

【0386】ステップS44において、制御部23は、ステップS43の処理で指定されたマーク点のRSPN\_ref\_EP\_startとoffset\_num\_pictures（図82）を取得する。

【0387】ステップS45において、制御部23は、ステップS44で取得したRSPN\_ref\_EP\_startに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0388】ステップS46において、制御部23は、AVデコーダ27を制御し、RSPN\_ref\_EP\_startで参照されるピクチャから（表示はしないで）、表示すべきピクチャをカウントアップしていき、カウント値がoffset\_num\_picturesになったとき、そのピクチャから表示を開始させる。

【0389】以上の処理を、図113乃至図115を参照して、さらに説明する。この例においては、DVRトランスポートストリームファイルは、ソースパケット番号Aからシーンが開始しており、ソースパケット番号BからソースパケットCまでCMが挿入されている。このため、図114に示されるように、EP\_mapには、RSPN\_EP\_startとしてのA、B、Cに対応して、PTS\_EP\_startとして、PTS(A)、PTS(B)、PTS(C)が登録されている。

【0390】また、図115に示されるように、シーンスタート、CMスタート、およびCMエンドのマークタイプに対応して、mark\_entryとrepresentative\_picture\_entryが登録されている。mark\_entryには、シーンスタート、CMスタート、およびCMエンドに対応して、RSPN\_ref\_EP\_startとして、それぞれA、B、Cが登録され、offset\_num\_picturesとして、M1、N1、N2が登録されている。同様に、representative\_picture\_entryには、RSPN\_ref\_EP\_startとして、シーンスタート、CMスタート、およびCMエンドに対応して、それぞれA、B、Cが登録され、offset\_num\_picturesとして、M2、N1、N2がそれぞれ登録されている。

【0391】シーンスタートに当たるピクチャから頭出して再生が指令された場合、パケット番号Aのデータか



ら始まるストリームからデコードが開始され、PTS(A)のピクチャから（表示をしないで）表示すべきピクチャをカウントアップをしていき、offset\_num\_picturesが、M1の値になったとき、そのピクチャから表示が開始される。

【0392】さらに、mark\_entryとrepresentative\_picture\_entryのシンタクスが、図82に示される構成である場合におけるCMスキップ再生の処理について、図116のフローチャートを参照して説明する。

【0393】ステップS61において、制御部23は、EP\_map（図70）、STC\_Info（図52）、Program\_Info（図54）、およびClipMark（図78）の情報を取得する。

【0394】ステップS62において、ユーザがCMスキップ再生を指令すると、ステップS63において、制御部23は、マークタイプ（図79）がCM開始点とCM終了点である各点のマーク情報として、RSPN\_ref\_EP\_STARTとoffset\_num\_pictures（図82）を取得する。そして、CM開始点のデータは、RSPN\_ref\_EP\_start(1),offset\_num\_pictures(1)とされ、CM終了点のデータは、RSPN\_ref\_EP\_start(2),offset\_num\_pictures(2)とされる。

【0395】ステップS64において、制御部23は、RSPN\_ref\_EP\_start(1),RSPN\_ref\_EP\_start(2)に対応するPTSをEP\_map（図70）から取得する。

【0396】ステップS65において、制御部23は、トランスポートストリームを記録媒体100から読み出させ、AVデコーダ27に供給させる。

【0397】ステップS66において、制御部23は、現在の表示画像がRSPN\_ref\_EP\_start(1)に対応するPTSのピクチャであるか否かを判定し、現在の表示画像がRSPN\_ref\_EP\_start(1)に対応するPTSのピクチャでない場合には、ステップS67に進み、ピクチャをそのまま継続的に表示させる。その後、処理はステップS65に戻り、それ以降の処理が繰り返し実行される。

【0398】ステップS66において、現在の表示画像がRSPN\_ref\_EP\_start(1)に対応するPTSのピクチャであると判定された場合、ステップS68に進み、制御部23は、AVデコーダ27を制御し、RSPN\_ref\_EP\_start(1)に対応するPTSのピクチャから表示するピクチャをカウントアップしていき、カウント値がoffset\_num\_pictures(1)になったとき、表示を停止させる。

【0399】ステップS69において、制御部23は、RSPN\_ref\_EP\_start(2)のソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0400】ステップS70において、制御部23は、AVデコーダ27を制御し、RSPN\_ref\_EP\_start(2)に対応するPTSのピクチャから（表示をしないで）表示すべきピクチャをカウントアップしていき、カウント値がoffset\_num\_pictures(2)になったとき、そのピクチャから表

示を開始させる。

【0401】以上の動作を、図113乃至図115を参照してさらに説明すると、まず、EP\_map（図114）をもとに、パケット番号B、Cに対応する時刻PTS(B),PTS(C)が得られる。そして、Clip AV streamがデコードされていき、表示時刻がPTS(B)になったとき、PTS(B)のピクチャから表示ピクチャがカウントアップされ、その値がN1（図115）になったとき、表示が停止される。

【0402】さらに、パケット番号Cのデータから始まるストリームからデコードが再開され、PTS(C)のピクチャから（表示をしないで）表示すべきピクチャをカウントアップしていき、その値がN2（図115）になったとき、そのピクチャから表示が再開される。

【0403】以上の処理は、CMスキップ再生に限らず、ClipMarkで指定された2点間のシーンをスキップさせて再生する場合にも、適用可能である。

【0404】次に、mark\_entryとrepresentative\_picture\_entryのシンタクスが、図84に示すような構成である場合における、マーク点で示されるシーンの頭出し再生処理について、図118のフローチャートを参照して説明する。

【0405】ステップS81において、EP\_map（図70）、STC\_Info（図52）、Program\_Info（図54）、並びにClipMark（図78）の情報が取得される。

【0406】ステップS82において、制御部23は、ClipMark（図78）のrepresentative\_picture\_entryまたはref\_thumbnail\_indexで参照されるピクチャからサムネイルのリストを生成し、GUIのメニュー画面として表示させる。ref\_thumbnail\_indexが有効な値を有する場合、representative\_picture\_entryよりref\_thumbnail\_indexが優先される。

【0407】ステップS83において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、メニュー画面上の中からユーザがサムネイル画像を選択し、そのサムネイルに対応づけられているマーク点を指定することで行われる。

【0408】ステップS84において、制御部23は、ユーザから指定されたmark\_entryのRSPN\_mark\_point（図84）を取得する。

【0409】ステップS85において、制御部23は、マーク点のRSPN\_mark\_pointより前にあり、かつ、最も近いエン트리ポイントのソースパケット番号を、EP\_map（図70）から取得する。

【0410】ステップS86において、制御部23は、ステップS85で取得したエン트리ポイントに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0411】ステップS87において、制御部23は、AVデコーダ27を制御し、RSPN\_mark\_pointで参照されるピクチャから表示を開始させる。

【0412】以上の処理を、図119乃至図121を参照してさらに説明する。この例においては、DVRトランスポートストリームファイルが、ソースパケットAでシーンスタートし、ソースパケット番号BからCまでCMが挿入されている。このため、図120のEP\_mapには、RSPN\_EP\_startとしてのA、B、Cに対応して、PTS\_EP\_startがそれぞれPTS(A)、PTS(B)、PTS(C)として登録されている。また、図121に示されるClipMarkに、シーンスタート、CMスタート、およびCMエンドに対応して、mark entryのRSPN\_mark\_pointとして、a1、b1、c1が、また、representative\_picture\_entryのRSPN\_mark\_pointとして、a2、b1、c1が、それぞれ登録されている。

【0413】シーンスタートにあたるピクチャから頭出して再生する場合、パケット番号A<a1とすると、パケット番号Aのデータから始まるストリームからデコードが開始され、ソースパケット番号a1に対応するピクチャから表示が開始される。

【0414】次に、mark\_entryとrepresentative\_picture\_entryのシンタクスが、図84に示されるような構成である場合におけるCMスキップ再生の処理について、図122と図123のフローチャートを参照して説明する。

【0415】ステップS101において、制御部23は、EP\_map(図70)、STC\_Info(図52)、Program\_Info(図54)、並びにClipMark(図70)の情報を取得する。

【0416】ステップS102において、ユーザは、CMスキップ再生を指定する。

【0417】ステップS103において、制御部23は、マークタイプ(図79)がCM開始点とCM終了点である各点のマーク情報のRSPN\_mark\_point(図84)を取得する。そして、制御部23は、CM開始点のデータをRSPN\_mark\_point(1)とし、CM終了点のデータをRSPN\_mark\_point(2)とする。

【0418】ステップS104において、制御部23は、記録媒体100からトランスポートストリームを読み出させ、AVデコーダ27に出力し、デコードさせる。

【0419】ステップS105において、制御部23は、現在の表示画像がRSPN\_mark\_point(1)に対応するピクチャであるか否かを判定し、現在の表示画像がRSPN\_mark\_point(1)に対応するピクチャでない場合には、ステップS106に進み、そのままピクチャを継続的に表示させる。その後、処理はステップS104に戻り、それ以降の処理が繰り返し実行される。

【0420】ステップS105において、現在の表示画像がRSPN\_mark\_point(1)に対応するピクチャであると判定された場合、ステップS107に進み、制御部23はAVデコーダ27を制御し、デコードおよび表示を停止させる。

【0421】次に、ステップS108において、RSPN\_mark\_point(2)より前にあり、かつ、最も近いエン트리ポイントのあるソースパケット番号がEP\_map(図70)から取得される。

【0422】ステップS109において、制御部23は、ステップS108で取得したエン트리ポイントに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0423】ステップS110において、制御部23は、AVデコーダ27を制御し、RSPN\_mark\_point(2)で参照されるピクチャから表示を再開させる。

【0424】以上の処理を図119乃至図121の例でさらに説明すると、Clip AV streamをデコードして行き、ソースパケット番号b1(図121)に対応する表示ピクチャになったとき、表示が停止される。そして、ソースパケット番号C<ソースパケット番号c1とすると、パケット番号Cのデータから始まるストリームからデコードが再開され、ソースパケット番号c1に対応するピクチャになったとき、そのピクチャから表示が再開される。

【0425】以上のようにして、図124に示されるように、PlayList上で、タイムスタンプにより所定の位置を指定し、このタイムスタンプを各ClipのClip Informationにおいて、データアドレスに変換し、Clip AV streamの所定の位置にアクセスすることができる。

【0426】より具体的には、図125に示されるように、PlayList上において、PlayListMarkとしてブックマークやリジューム点を、ユーザが時間軸上のタイムスタンプとして指定すると、そのPlayListは再生するとき、そのPlayListが参照しているClipのClipMarkを使用して、Clip AV streamのシーン開始点やシーン終了点にアクセスすることができる。

【0427】なお、ClipMarkのシンタクスは、図78の例に替えて、図126に示すようにすることもできる。

【0428】この例においては、RSPN\_markが、図78のreserved\_for\_MakerID、mark\_entry()、およびrepresentative\_picture\_entry()に替えて挿入されている。このRSPN\_markの32ビットのフィールドは、AVストリームファイル上で、そのマークが参照するアクセスユニットの第1バイト目を含むソースパケットの相対アドレスを示す。RSPN\_markは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClip Information fileにおいて定義され、offset\_SPNの値を初期値としてカウントされる。

【0429】その他の構成は、図78における場合と同様である。

【0430】ClipMarkのシンタクスは、さらに図127に示すように構成することもできる。この例において

は、図126におけるRSPN\_markの代わりに、RSPN\_ref\_EP\_startとoffset\_num\_picturesが挿入されている。これらは、図82に示した場合と同様のものである。

【0431】次に、図128は、ClipInfo()のシンタクスの別例を示す。

【0432】Clip\_service\_typeは、AVストリームファイルのタイプを示す。例えば、Clip\_service\_typeは、ビデオレコーディングやオーディオレコーディングなどのタイプを示す。また、例えば、Clip\_service\_typeは、デジタルTV放送のプログラムが示すサービスタイプと同じ意味を持たせても良い。例えば、日本のデジタルBS放送の場合、サービスタイプは、テレビサービス、音声サービスおよびデータ放送サービスの3種類を持つ。AVストリームが含むプログラムのサービスタイプを代表する値をClip\_service\_typeにセットする。

【0433】transcode\_mode\_flagは、デジタル放送から受信されたMPEG2 トランスポートストリームの記録方法を示すフラグである。このフラグが1にセットされている場合、Clipに対応するAVストリームファイル中の少なくとも1つのエレメンタリーストリームは再符号化されて記録されたことを示す。このフラグが1にセットされている場合、Clipに対応するAVストリームファイル中のすべてのエレメンタリーストリームはデジタル放送から受信されたままの内容で何も変更されずに記録されたことを示す。

【0434】その他のシンタクスフィールドは、図46で説明した同名のフィールドと同じ意味を持つ。

【0435】次に、図129を参照して、ProgramInfo()の別例について説明する。

【0436】AVストリームファイルの中で本フォーマットが規定するところのプログラム内容が一定であるソースパケット列を、program-sequenceと呼ぶ。

【0437】AVストリームファイルの中で、新しいprogram-sequenceが開始するアドレスをProgramInfo()にストアする。このアドレスは、SPN\_program\_sequence\_startにより示される。

【0438】AVストリームファイルの中にある最後のprogram-sequence以外のprogram-sequenceは、そのSPN\_program\_sequence\_startで指されるソースパケットから開始し、その次のSPN\_program\_sequence\_startで指されるソースパケットの直前のソースパケットで終了する。最後のprogram-sequenceは、そのSPN\_program\_sequence\_startで指されるソースパケットから開始し、AVストリームファイルの最後のソースパケットで終了する。

【0439】program-sequenceは、STC-sequenceの境界をまたいで良い。

【0440】lengthは、このlengthフィールドの直後のバイトからProgramInfo()の最後のバイトまでのバイト数を示す。

【0441】num\_of\_program\_sequencesは、AVストリー

ムファイルの中にあるprogram-sequenceの数を示す。

【0442】SPN\_program\_sequence\_startは、AVストリームファイル上でprogram-sequenceが開始するアドレスを示す。SPN\_program\_sequence\_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットから、ゼロを初期値としてカウントされる。

【0443】ProgramInfo()の中でエントリーされるSPN\_program\_sequence\_startの値は、昇順に並んでいる。

【0444】SPN\_program\_sequence\_startは、そのprogram-sequenceに対する最初のPMTを持つソースパケットを指していることを前提とする。SPN\_program\_sequence\_startは、記録機(図1の記録再生装置1)がトランスポートストリーム中のPSI/SIを解析することによって作られる。記録機がPSI/SIを解析し、その変化を検出するまでの遅延時間が必要なために、SPN\_program\_sequence\_startは、実際のPSI/SIの変化点から所定の時間以内にあるソースパケットを指しても良い。

【0445】program\_map\_PIDは、そのprogram-sequenceに適用できるPMT(program map table)を持つトランスポートパケットのPIDの値である。

【0446】num\_of\_streams\_in\_psiは、そのprogram-sequenceの中で定義されるエレメンタリーストリームの数を示す。

【0447】num\_of\_groupsは、そのprogram-sequenceの中で定義されるエレメンタリーストリームのグループの数を示す。num\_of\_groupsは、1以上の値である。

【0448】トランスポートストリームのPSI/SIがエレメンタリーストリームのグループ情報を持つ場合、num\_of\_groupsは、1以上の値をとることを想定している。それぞれのグループは、マルチ・ビュー・プログラム中の1つのビューを構成する。

【0449】stream\_PIDは、そのprogram-sequenceのprogram\_map\_PIDが参照するところのPMTの中で定義されているエレメンタリーストリームに対するPIDの値を示す。

【0450】StreamCodingInfo()は、上記stream\_PIDで指されるエレメンタリーストリームの情報を示す。詳細は後述する。

【0451】num\_of\_streams\_in\_groupは、エレメンタリーストリームのグループが持つエレメンタリーストリームの数を示す。

【0452】stream\_indexは、上記エレメンタリーストリームのグループが持つエレメンタリーストリームに対応するところの、シンタクス中のstream\_indexのfor-loopで記述される順番で定義されるstream\_indexの値を示す。

【0453】図30は、StreamCodingInfo()のシンタクスを示す。

【0454】lengthは、このlengthフィールドの直後の

バイトからStreamCodingInfo()の最後のバイトまでのバイト数を示す。

【0455】stream\_coding\_typeは、このStreamCodingInfo()に対応するstream\_PIDで指されるエレメンタリーストリームの符号化タイプを示す。値の意味は、図131に示される。

【0456】video\_formatは、このStreamCodingInfo()に対応するstream\_PIDで指されるビデオストリートのビデオフォーマットを示す。

【0457】値の意味は、図56と同じである。

【0458】frame\_rateは、このStreamCodingInfo()に対応するstream\_PIDで指されるビデオストリートのフレームレートを示す。

【0459】値の意味は、図57と同じである。

【0460】display\_aspect\_ratioは、このStreamCodingInfo()に対応するstream\_PIDで指されるビデオストリートのディスプレイ・アスペクト・レシオを示す。

【0461】値の意味は、図58と同じである。

【0462】cc\_flagは、このStreamCodingInfo()に対応するstream\_PIDで指されるビデオストリートのクロード・キャプション(closed caption data)信号が符号化されているか否かを示すフラグである。

【0463】original\_video\_format\_flagは、このStreamCodingInfo()の中にoriginal\_video\_formatとoriginal\_display\_aspect\_ratioが存在するか否かを示すフラグである。

【0464】original\_video\_formatは、このStreamCodingInfo()に対応するstream\_PIDで指されるビデオストリートの符号化される前のオリジナルのビデオフォーマットである。値の意味は、上記のvideo\_format(図56)と同じである。

【0465】original\_display\_aspect\_ratioは、このStreamCodingInfo()に対応するstream\_PIDで指されるビデオストリートの符号化される前のオリジナルのディスプレイ・アスペクト・レシオである。値の意味は、上記のdisplay\_aspect\_ratio(図58)と同じである。

【0466】ビデオストリートの共にマルチメディアデータストリートの(BMLストリートの、字幕など)が多重化されているトランスポートストリートをトランス・コーディングする場合において、ビデオストリートの再エンコードされることによって、そのビデオフォーマットが変化し(例えば、1080iから480iへ変化する)、一方、マルチメディアデータストリートのオリジナルのストリートのままの内容を保つ場合を考える。この時、新しいビデオストリートのマルチメディアデータストリートの間に情報のミスマッチが生じる場合がある。例えば、マルチメディアデータストリートの表示に関するパラメータは、オリジナルのビデオストリートのビデオフォーマットを想定して決められているにもかかわらず、ビデオストリートの再エンコードによって、そのビデオ

フォーマットが変化した場合である。このような場合、original\_video\_formatとoriginal\_display\_aspect\_ratioに、オリジナルのビデオストリートのに関する情報を保存する。再生機(図1の記録再生装置1)は、上記の新しいビデオストリートのとマルチメディアデータストリートのから次のようにして、表示画像をつくる。

【0467】・ビデオストリートの、original\_video\_formatとoriginal\_display\_aspect\_ratioで示されるビデオフォーマットにアップ・サンプリングされる。・そのアップ・サンプリングされた画像とマルチメディアデータストリートのが合成されて、正しい表示画像をつくる。

【0468】audio\_presentation\_typeは、このStreamCodingInfo()に対応するstream\_PIDで指されるオーディオストリートのプレゼンテーション・タイプを示す。

【0469】値の意味は、図61のaudio\_component\_typeと同じである。

【0470】sampling\_frequencyは、このStreamCodingInfo()に対応するstream\_PIDで指されるオーディオストリートのサンプリング周波数を示す。

【0471】値の意味は、図62と同じである。

【0472】次に、EP\_mapの別例を示す。このEP\_mapは、図70と図72で説明したPTS\_EP\_startとRSPN\_EP\_startのデータをそのままEP\_mapにストアするのではなく、データ量の削減のために圧縮符号化して(図1の制御部23が実行する)EP\_mapにストアする。

【0473】EP\_mapは、少なくとも1つ以上のEP\_map\_for\_one\_stream\_PID()と呼ばれるサブテーブルから成る。サブテーブルEP\_map\_for\_one\_stream\_PID()は、同じPIDの値を持つトランスポートパケットによって伝送される1つのエレメンタリーストリートに対して作られる。

【0474】EP\_mapは、すべてのEP\_map\_for\_one\_stream\_PID()のための次の情報を持つ。

【0475】(1) stream\_PID:そのEP\_map\_for\_one\_stream\_PID()によって指されるエレメンタリーストリートを伝送するところのトランスポートパケットのPIDを特定する。

(2) EP\_stream\_type:そのEP\_map\_for\_one\_stream\_PID()のエントリーポイントのタイプを示す。

(3) EP\_map\_for\_one\_stream\_PID()のエントリー。それぞれのエントリーは次の情報を持つ:

・PTS\_EP\_start: エントリーポイントにおけるアクセスユニットのPTSの値を示す。

・RSPN\_EP\_start: AVストリートの中で、そのエントリーポイントのPTS\_EP\_startによって参照されるところのアクセスユニットの開始するアドレスを示す。

【0476】EP\_mapが1つ以上のエレメンタリーストリートを指す場合、EP\_mapは1つ以上のEP\_map\_for\_one\_stream\_PID()サブテーブルを持っても良い。図67は、EP\_mapがClip中の3個のビデオストリートを指している

場合のEP\_mapの例を示している。

【0477】1つのEP\_map\_for\_one\_stream\_PID()は、STCの不連続点に関係なく1つのテーブルに作られる。

【0478】RSPN\_EP\_startの値とSTC\_Info()において定義されるRSPN\_STC\_startの値を比較する事により、EP\_mapの中にあるSTC\_sequenceの境界が分かる(図68参照)。

【0479】EP\_mapは、同じPIDで伝送されるストリームの連続した範囲に対して、1つのEP\_map\_for\_one\_stream\_PIDを持つ。

【0480】図69の場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP\_map\_for\_one\_stream\_PIDを持たねばならない。

【0481】EP\_map\_for\_one\_stream\_PID()のテーブルのデータサイズを削減し、かつデータサーチのパフォーマンスを改善するために、EP\_map\_for\_one\_stream\_PID()は2つのサブテーブル、すなわち、EP\_coarseとEP\_fineに分けられる(図132参照)。

【0482】EP\_fineエントリーは、PTS\_EP\_start および RSPN\_EP\_startのLSB(Least Significant Bit)側のビット情報を持つ(図133と図134を参照)。EP\_coarseエントリーは、PTS\_EP\_start および RSPN\_EP\_startのMSB(Most Significant Bit)側のビット情報とそれに対応するEP\_fineのテーブル中のエントリー番号(同じPTS\_EP\_startから取り出したLSB側のビットを持つEP\_fineテーブルの中のエントリー)を持つ。

【0483】EP\_coarseサブテーブルの中のエントリー数は、EP\_fineサブテーブルのエントリー数に比べてかなり小さい。次の場合に、EP\_coarseのエントリーが作られる：

- ・最初のPTS\_EP\_fineのエントリー
- ・PTS\_EP\_fineの値がラップ・アラウンドした後の最初のPTS\_EP\_fineのエントリー
- ・RSPN\_EP\_fineの値がラップ・アラウンドした後の最初のRSPN\_EP\_fineのエントリー(図135参照)。

【0484】EP\_mapを使用する場合のAVストリームへのランダムアクセスの例について説明する。

【0485】あるPlayListのグローバルな時間軸上で30分後から、その再生を開始したいとする。

・最初にPlayListの中のグローバルな時間軸上で30分後に対応するところ時刻を含むPlayItemのSTC-sequence-idを調べる。

・前記STC-sequenceのローカルな時間軸上で30分後に対応するところのPTSの値を導く。

・前記STC-sequenceのRSPN\_STC\_startをSTC\_Infoから導く。

・EP\_coarse サブテーブルの中で、RSPN\_EP\_coarseが前記RSPN\_STC\_start以上であるところのエントリーから、データサーチが開始される。EP\_coarse サブテーブルの

中で、前記必要なPTSに最も近く、かつその値より時間的に前の値を持つPTS\_EP\_coarseのエントリーを見つける。

・EP\_fineサブテーブルの中で、前記見つけたPTS\_EP\_coarseに対応するところのEP\_fineのエントリー番号から、データサーチが開始される。前記必要なPTSに最も近く、かつその値より時間的に前の値を持つPTS\_EP\_fineを持つエントリーを見つける。

・前記必要なPTSのアクセスユニットの読み出し開始のためのRSPN\_EP\_startは、前記見つけたPTS\_EP\_coarseに対応するRSPN\_EP\_coarseのLSB 18ビットを前記見つけたPTS\_EP\_fineに対応するRSPN\_EP\_fineのビットで置きかえることにより決定される。

【0486】上記説明したEP\_mapのシンタクスを図136に示す。

【0487】number\_of\_stream\_PID\_entriesは、EP\_mapの中でのEP\_map\_for\_one\_stream\_PIDのエントリー数を示す。

【0488】stream\_PID[k]は、EP\_mapの中でk番目にエントリーされるEP\_map\_for\_one\_stream\_PIDによって参照されるエレメンタリーストリームを伝送するところのトランスポートパケットのPIDの値を示す。

【0489】EP\_stream\_type[k]は、前記EP\_map\_for\_one\_stream\_PIDのによって参照されるエレメンタリーストリームのタイプを示す。値の意味を図137の表に示す。

【0490】EP\_stream\_type[k]が0または1の場合、そのエレメンタリーストリームはビデオストリームである。Video type 1 と video type 2 の意味についてはEP\_video\_type(図139)の説明のところで後述する。

【0491】EP\_stream\_type[k]が2の場合、そのエレメンタリーストリームはオーディオストリームである。

【0492】num\_EP\_coarse\_entries[k]は、前記EP\_map\_for\_one\_stream\_PIDの中にあるEP-coarseエントリーの数を示す。

【0493】num\_EP\_fine\_entries[k]は、前記EP\_map\_for\_one\_stream\_PIDの中にあるEP-fineエントリーの数を示す。

【0494】EP\_map\_for\_one\_stream\_PID\_start\_address[k]は、EP\_map()の中で前記EP\_map\_for\_one\_stream\_PIDが始まる相対バイト位置を示す。この値は、EP\_map()の第1バイト目からのバイト数で示される。

【0495】EP\_map\_for\_one\_stream\_PIDのシンタクスを図138に示す。これのセマンティクスを説明するために、EP\_map\_for\_one\_stream\_PIDにストアされるデータの元となるところのPTS\_EP\_start とRSPN\_EP\_startの意味について説明する。

【0496】PTS\_EP\_startとそれに関連付けられたRSPN\_EP\_startは、AVストリーム上のエントリーポイントを

指す。そして、PTS\_EP\_fineとそれに関連付けられたPTS\_EP\_coarseは、同一のPTS\_EP\_startから導かれる。また、RSPN\_EP\_fineとそれに関連付けられたRSPN\_EP\_coarseは同じRSPN\_EP\_startから導かれる。PTS\_EP\_start と RSPN\_EP\_startは次のように定義される。

【0497】PTS\_EP\_startは、33-bit長の符号なし整数である。PTS\_EP\_startの定義は、EP\_map\_for\_one\_stream\_PIDに対するEP\_stream\_typeの値により異なる。

【0498】EP\_stream\_typeがゼロであるとき('video type 1')、PTS\_EP\_startは、AVストリームの中でシーケンスヘッダから開始するビデオアクセスユニットの33ビット長のPTSを示す。

【0499】EP\_stream\_typeが2であるとき('audio')、PTS\_EP\_startは、AVストリームの中でシーケンスヘッダから開始するビデオアクセスユニットの33ビット長のPTSを示す。

【0500】EP\_stream\_typeが1であるとき('video type 2')、PTS\_EP\_startは、それに関連づけられたEP\_video\_typeの値に従って、図139に定義されるところのビデオアクセスユニットの33ビット長のPTSを示す。

【0501】RSPN\_EP\_startは、32ビットの符号なし整数である。RSPN\_EP\_startの定義はEP\_map\_for\_one\_stream\_PIDに対するEP\_stream\_typeの値により異なる。

【0502】EP\_stream\_typeがゼロであるとき('video type 1')、このフィールドはPTS\_EP\_startに関連付けられたビデオアクセスユニットの第1バイト目を含むソースパケットのAVストリームの中でのアドレスを示す。

【0503】EP\_stream\_typeが2であるとき('audio')、このフィールドはPTS\_EP\_startに関連付けられたオーディオアクセスユニットの第1バイト目を含むソースパケットのAVストリームの中でのアドレスを示す。

【0504】EP\_stream\_typeが1であるとき('video type 2')、RSPN\_EP\_startの意味は、それに関連づけられたEP\_video\_typeの値に従って、図139に定義されるものである。

【0505】RSPN\_EP\_startは、ソースパケット番号の単位で表され、それはAVストリームファイルの中の最初のソースパケットからゼロを初期値としてカウントされる。

【0506】EP\_map\_for\_one\_stream\_PIDのセマンティクスを説明する。

【0507】EP\_fine\_table\_start\_addressは、最初のEP\_video\_type[EP\_fine\_id]の第1バイト目の開始アドレスを示し、それはEP\_map\_for\_one\_stream\_PID()の第1バイト目からの相対バイト数で表される。相対バイト数はゼロから開始する。

【0508】ref\_to\_EP\_fine\_idは、このフィールドに続くPTS\_EP\_coarseに関連づけられるところのPTS\_EP\_fi

neを持つEP\_fineエントリーの番号を示す。PTS\_EP\_fineとそれに関連付けられるPTS\_EP\_coarseは、同じPTS\_EP\_startから導かれる。

【0509】ref\_to\_EP\_fine\_idは、EP\_fine\_idのfor-loopの中で記述される順番で定義されるところのEP\_fine\_idの値により与えられる。

【0510】PTS\_EP\_coarseとPTS\_EP\_fine、そしてRSPN\_EP\_coarseとRSPN\_EP\_fineは、次のように導かれる。

【0511】EP\_fineサブテーブルに、Nf個のエントリーがあり、これらのエントリーはそれらに関連するRSPN\_EP\_startの値の昇順に並んでいるとする。

【0512】それぞれのPTS\_EP\_fineエントリーは、そのPTS\_EP\_startから次のように導かれる。

【0513】 $PTS\_EP\_fine[EP\_fine\_id] = (PTS\_EP\_start[EP\_fine\_id] \gg 9) \% 2^{12}$

【0514】PTS\_EP\_coarse とそのPTS\_EP\_fineの関係は次のとおりである。

【0515】 $PTS\_EP\_coarse[i] = (PTS\_EP\_start[ref\_to\_EP\_fine\_id[i]] \gg 19) \% 2^{14}$

20  $TS\_EP\_fine[ref\_to\_EP\_fine\_id[i]] = (PTS\_EP\_start[ref\_to\_EP\_fine\_id[i]] \gg 9) \% 2^{12}$

【0516】それぞれのRSPN\_EP\_fineエントリーは、そのRSPN\_EP\_startから次のように導かれる。

【0517】 $RSPN\_EP\_fine[EP\_fine\_id] = RSPN\_EP\_start[EP\_fine\_id] \% 2^{18}$

【0518】RSPN\_EP\_coarse とその RSPN\_EP\_fineの関係は次のとおりである。

【0519】 $RSPN\_EP\_coarse[i] = RSPN\_EP\_start[ref\_to\_EP\_fine\_id[i]]$

30  $RSPN\_EP\_fine[ref\_to\_EP\_fine\_id[i]] = RSPN\_EP\_start[ref\_to\_EP\_fine\_id[i]] \% 2^{18}$

【0520】EP\_video\_type[EP\_fine\_id]は、EP\_stream\_typeが1('video type 2')である場合、PTS\_EP\_fine[EP\_fine\_id]およびRSPN\_EP\_fine[EP\_fine\_id]が指すところのビデオアクセスユニットのタイプを示す。値の意味は、図139に示すとおりである。

【0521】EP\_map\_for\_one\_stream\_PID()は次の制限を満たす。

【0522】・EP\_map\_for\_one\_stream\_PID()の中でRSPN\_EP\_coarseのエントリーは、RSPN\_EP\_coarseの値の昇順で並んでいなければならない。

・EP\_map\_for\_one\_stream\_PID()の中でRSPN\_EP\_fineのエントリーは、それに関連するRSPN\_EP\_startの値の昇順で並んでいなければならない。

・次の場合に、PTS\_EP\_coarseとRSPN\_EP\_coarseのエントリーが作られる：

・・最初のPTS\_EP\_fineのエントリー

・・PTS\_EP\_fineの値がラップ・アラウンドした後の最初のPTS\_EP\_fineのエントリー

・・RSPN\_EP\_fineの値がラップ・アラウンドした後の最

初のRSPN\_EP\_fineのエントリー。

【0523】次に、図140はClip AVストリームファイルとそれに関連するClip Informationファイルの記録動作のフローチャートを示す。図1の記録再生装置を参照して説明する。

【0524】ステップS201で、制御部23は、端子11および12から入力されるAV入力をエンコードして得たトランスポートストリーム、または端子13のデジタルインタフェースから入力されるトランスポートストリームをファイル化して、Clip AVストリームファイルを作成して記録する。

【0525】ステップS202で、制御部23は上記AVストリームファイルについてのClipInfoを作成する。

【0526】ステップS203で、制御部23は上記AVストリームファイルについてのSTC\_Infoを作成する。

【0527】ステップS204で、制御部23は上記AVストリームファイルについてのProgram\_Infoを作成する。

【0528】ステップS205で、制御部23は上記AVストリームファイルについてのCPI(EP-mapまたはTU-map)を作成する。

【0529】ステップS206で、制御部23は上記AVストリームファイルについてのClipMarkを作成する。

【0530】ステップS207で、制御部23は上記ClipInfo、STC\_Info、ProgramInfo、CPI、およびClipMarkがストアされたClip Informationファイルを記録する。

【0531】なお、ここでは各処理を時系列に説明したが、ステップS201からステップS206は実際には同時に動作するものである。

【0532】次に、STC\_Infoの作成の動作例を図14のフローチャートを用いて説明する。この処理は図1の多重化ストリーム解析部18で行われる。

【0533】ステップS221において、ストリーム解析部18はPCRパケット受信されたかどうかを調べる。ステップS221において、Noの場合はステップS221へ戻り、Yesの場合はステップS222へ進む。

【0534】ステップS222において、STCの不連続が検出されたかを調べる。Noの場合は、ステップS221へ戻る。Yesの場合は、ステップS223へ進む。なお、記録開始後、最初に受信されたPCRパケットの場合は、必ずステップS223へ進む。

【0535】ステップS223において、新しいSTCの最初のPCRを送信するトランスポートパケットの番号(アドレス)を取得する。

【0536】ステップS224において、STC\_Infoを作成する。

【0537】ステップS225において、最後のトランスポートパケットが入力終了したかどうかを調べる。Noの場合は、ステップS221へ戻り、Yesの場合は処理を終了する。

【0538】Program\_Infoの作成の動作例を図142のフローチャートを用いて説明する。この処理は図1の多重化ストリーム解析部18で行われる。

【0539】ステップS241において、ストリーム解析部18はPSI/SIを含むトランスポートパケットが受信されたかどうかを調べる。ここで、PSI/SIのトランスポートパケットは、具体的には、PAT、PMT、SITのパケットである。SITは、DVB規格で規定されているパースシャルトランスポートストリームのサービス情報が記述されているトランスポートパケットである。ステップ241において、Noの場合はステップS241へ戻り、Yesの場合はステップS242へ進む。

【0540】ステップS242において、PSI/SIの内容が変わったかを調べる。すなわち、PAT、PMT、SITのそれぞれの内容が、以前に受信したそれぞれの内容と比べて変化したかどうかを調べる。内容が変化していない場合は、ステップS241へ戻る。内容が変化した場合は、ステップS243へ進む。なお、記録開始後、最初に受信されたPSI/SIの場合は、必ずステップS243へ進む。

【0541】ステップS243において、新しいPSI/SIを送信するトランスポートパケットの番号(アドレス)とその内容を取得する。

【0542】ステップS244において、Program-sequenceの情報を作成する。

【0543】ステップS245において、最後のトランスポートパケットが入力終了したかどうかを調べる。Noの場合は、ステップS241へ戻り、Yesの場合は処理を終了する。

【0544】次にEP\_mapの作成の動作例を図143のフローチャートを用いて説明する。この処理は図1の多重化ストリーム解析部18で行われる。

【0545】ステップS261でストリーム解析部18は、記録するAVプログラムのビデオのPIDをセットする。トランスポートストリームの中に複数のビデオが含まれている場合は、それぞれのビデオPIDをセットする。

【0546】ステップS262でストリーム解析部18は、ビデオのトランスポートパケットを受信する。

【0547】ステップS263でストリーム解析部は、トランスポートパケットのペイロード(パケットヘッダーに続くデータ部)がPESパケットの第1バイト目から開始しているかを調べる(PESパケットは、MPEG2で規定されているパケットであり、エレメンタリーストリームをパケット化するものである)。これは、トランスポートパケットヘッダにある"payload\_unit\_start\_indicator"の値を調べることによりわかり、この値が1である場合、トランスポートパケットのペイロードがPESパケットの第1バイト目から開始する。ステップS263でNoの場合は、ステップ262へ戻り、Yesの場合は、ス

ステップS264へ進む。

【0548】ステップS264でストリーム解析部は、PESパケットのペイロードが、MPEGビデオのsequence\_header\_code(32ビット長で"0x000001B3"の符号)の第1バイト目から開始しているかを調べる。ステップS264でNoの場合は、ステップS262へ戻り、Yesの場合は、ステップS265へ進む。

【0549】ステップS265へ進んだ場合、現在のトランスポート packets をエンタリーポイントとする。

【0550】ステップS266でストリーム解析部は、上記パケットの packet 番号と上記sequence\_header\_code から開始する1ピクチャのPTSとそのエンタリーポイントが属するビデオのPIDを取得し、制御部23へ入力する。制御部23はEP\_mapを作成する。

【0551】ステップS267で、現在のパケットが最後に入力されるトランスポート packets であるかどうかを判定する。最後の packets でない場合、ステップS262へ戻る。最後の packets である場合、処理を終了する。

【0552】図144は、アナログAV信号をエンコードして記録する場合、図81に示すシンタクスのClipMarkの作成方法を説明するフローチャートを示す。

【0553】ステップS281で、解析部14は端子11, 12からの入力AV信号を解析して特徴点を検出する。特徴点はAVストリームの内容に起因する特徴的なシーンを指定する、例えば番組の頭だし点やシーンチェンジ点などである。

【0554】ステップS282で、制御部23は特徴点の画像のPTSを取得する。

【0555】ステップS283で、制御部23は特徴点の情報をClipMarkにストアする。具体的には、本実施例のClipMarkのシンタクスとセマンティクスで説明した情報をストアする。

【0556】図145は、デジタルインタフェースから入力されたトランスポートストリームを記録する場合、図81に示すシンタクスのClipMarkの作成方法を説明するフローチャートを示す。

【0557】ステップS301で、デマルチプレクサ26および制御部23は記録するプログラムのエレメンタリストリームPIDの取得する。解析対象のエレメンタリストリームが複数ある場合は、すべてのエレメンタリストリームPIDを取得する。

【0558】ステップS302で、デマルチプレクサ26は、端子13から入力されるトランスポートストリームのプログラムからエレメンタリストリームを分離し、それをAVデコーダ27へ入力する。AVデコーダ27は、入力ストリームをデコードしてAV信号を出力する。

【0559】ステップS303で、解析部14は上記AV信号を解析して特徴点を検出する。

【0560】ステップS304で、制御部23は特徴点

の画像のPTSとそれが属するSTCのSTC-sequence-idを取得する。

【0561】ステップS305で、制御部23は特徴点の情報をClipMarkにストアする。具体的には、本実施例のClipMarkのシンタクスとセマンティクスで説明した情報をストアする。

【0562】次に、EP\_mapを使用して特殊再生を行う方法を説明する。EP\_mapはランダムアクセス再生のために役立つ。

【0563】デジタル放送のトランスポートストリームの中では、ビデオPIDが変化する場合があるので、デコーダは記録されているトランスポートストリームの中のPIDマッピングを知る必要がある。そのために、EP\_mapはEP\_map\_for\_one\_stream\_PID()と呼ばれるサブテーブル毎にそれが参照するビデオPIDの値を持ち、また、ProgramInfoはPIDマッピングについての情報を持つ。

【0564】図146は、トランスポートストリームの中でビデオPIDの値が変化するDVR MPEG2 TSの例を示す。この場合、EP\_mapはビデオPID毎にサブテーブルを持つ。

【0565】図147は、1ピクチャサーチ（トリックプレイ、チャプターサーチなど）をする場合のプレーヤモデルを示す。1ピクチャサーチは次のステップで行う。

【0566】1) 最初に、ファイルシステム112はディスク111（図1の記録媒体100に対応する）からClip Information file(EP\_map, STC\_Info, ProgramInfo)のデータを読む、そのデータはホストコントローラ115へ送られる。

2) ユーザインタフェースは、再生するプログラム番号およびサーチ開始時間のPTSをセットする。そしてその値はホストコントローラ115へ送られる。

3) ホストコントローラ115は、サーチ開始時間に対応するところのRSPN\_EP\_startが指すソースパケットのビデオPIDをデマルチプレクサ113にセットする。

4) ホストコントローラ115は、前記RSPN\_EP\_startのソースパケット番号に対応するデータアドレスをファイルシステム112にセットする。

5) ファイルシステム112は、指定されたデータアドレスからDVR MPEG2 トランスポートストリームを読み出す。

6) ユーザが次のサーチ時間をセットする場合は、上記ステップ2)へ戻る。

【0567】次に、図148は、オリジナルのAVストリームファイルと、そのストリームの部分的な再生範囲のストリームを消去する編集を行った後のAVストリームファイルの例を示す。

【0568】編集前に、Virtual Playlistは、オリジナルAVストリーム上のIN\_timeとOUT\_timeを指しているとする。この時、Virtual Playlistが使用していないスト



リーム部分を消去する編集（ミニマイズ編集）をした場合、それはオリジナルAVストリームを図148に示す編集後のストリームへ変える。オリジナルAVストリームの先頭からX点までのデータと、Y点から最後までまでのデータが消去される。以下の説明では、このX点とY点を決める方法の例を説明する。

【0569】図149は、AVストリームの内容を解析することをしないで、IN点の前の不要なデータを消去する方法を説明する図である。PlayListはオリジナルAVストリーム上のIN点を指す。また、そのAVストリームのEP\_mapを図示する。IN点が指すピクチャをデコードするためには、アドレスISA2から開始するIピクチャが必要である。

【0570】また、X点の後で、PAT、PMTおよびPCRパケットが必要である。RSPN\_EP\_start=ISA1のPTSはpts1であり、RSPN\_EP\_start=ISA2のPTSはpts2である。pts1とpts2のシステムタイムベースの時間差が100 msec以上ならば、アドレスISA1とISA2の間にはPAT、PMTおよびPCRパケットが存在する（少なくとも、SESF、DV B、ATSC、ISDBの場合はそうである）。

【0571】したがって、X点はアドレスISA1の前に決められる。そして、X点はアラインドユニットの境界でなければならない。

【0572】記録装置（図1の記録再生装置1）は、AVストリームの内容を解析することをしないで、X点をEP\_mapを使用して次のステップで決めることができる。

【0573】1）システムタイムベース上でIN timeのPTSに最も近く、かつそれよりも過去の表示時刻のPTSの値を持つRSPN\_EP\_startを見つける。

2）ステップ1）で見つけたRSPN\_EP\_startのPTSの値よりも少なくとも100 msec過去の表示時刻のPTSの値を持つRSPN\_EP\_startを見つける。

3）X点は、ステップ2）で見つけたRSPN\_EP\_startよりも前に決められる。そして、X点はアラインドユニットの境界でなければならない。

【0574】この方法は、X点を決めるためにAVストリームのデータを読み出し、その内容を解析することを必要としないので、簡単である。しかし、編集後のAVストリームは、そのPlayListの再生には不要なデータを残してしまう場合がある。もし、X点を決めるためにAVストリームのデータを読み出し、その内容を解析するならば、そのPlayListの再生には不要なデータをより効率良く消去できる。

【0575】図150は、AVストリームの内容を解析することをしないで、OUT点の後ろの不要なデータを消去する方法を説明する図である。PlayListはオリジナルAVストリーム上のOUT点を指す。また、そのAVストリームのEP\_mapを図示する。

【0576】RSPN\_EP\_start=ISA4から開始するビデオシーケンスは次に示すものであることを前提とする。

I2 B0 B1 P5 . . .

ここで、I,P,BはそれぞれIピクチャ、PピクチャそしてBピクチャを表す。数字は表示順序を表す。この処理において、記録装置がAVストリームの内容を解析しない場合、記録装置はOUT\_timeのPTSが参照するところのピクチャの情報（ピクチャコーディングタイプ、テンポラル・レファレンスなど）がわからない。OUT\_timeのPTSはピクチャB0またはB1を参照しているかもしれない（記録装置がAVストリームの内容を解析しない場合、このことはわからない）、この場合、ピクチャB0,B1をデコードするためにはI2が必要である。I2のPTSはOUT timeのPTSよりも大きい（OUT\_time < pts4、ここでpts4はI2のPTSである）。I2のPTSはOUT\_timeのPTSよりも大きい、B0、B1のためにI2が必要である。

【0577】したがって、Y点は図に示すアドレスISA5の後ろに決められる。ISA5は、EP\_mapの中でISA4の直後にあるRSPN\_EP\_startの値である。Y点はまたアラインドユニットの境界でなければならない。

【0578】記録装置は、AVストリームの内容を解析することをしないで、Y点をEP\_mapを使用して次のステップで決めることができる。

【0579】1）システムタイムベース上でOUT timeのPTSに最も近く、かつそれよりも未来の表示時刻のPTSの値を持つRSPN\_EP\_startを見つける。

2）ステップ1）で見つけたRSPN\_EP\_startの直後にあるRSPN\_EP\_startを見つける。

3）Y点は、ステップ2）で見つけたRSPN\_EP\_startよりも後ろに決められる。そして、Y点はアラインドユニットの境界でなければならない。

【0580】この方法は、Y点を決めるためにAVストリームのデータを読み出し、その内容を解析することを必要としないので、簡単である。しかし、編集後のAVストリームは、そのPlayListの再生には不要なデータを残してしまう場合がある。もし、Y点を決めるためにAVストリームのデータを読み出し、その内容を解析するならば、そのPlayListの再生には不要なデータをより効率良く消去できる。

【0581】このようなシンタクス、データ構造、規則に基づく事により、記録媒体に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

【0582】本発明のデータベース構成によれば、PlayListファイルやClip Informationファイルを別々に分離して記録するので、編集などによって、あるPlayListやClipの内容を変更したとき、そのファイルに関係のない他のファイルを変更する必要がない。したがって、ファイルの内容の変更が容易に行え、またその変更および記録にかかる時間を小さくできる。

【0583】また、最初にInfo.dvrだけを読み出して、ディスクの記録内容をユーザインタフェースへ提示し、ユーザが再生指示したPlayListファイルとそれに関連するClip Informationファイルだけをディスクから読み出すようにすれば、ユーザの待ち時間を小さくすることができる。

【0584】もし、すべてのPlayListファイルやClip Informationファイルを1つのファイルにまとめて記録すると、そのファイルサイズは非常に大きくなる。そのために、そのファイルの内容を変更して、それを記録するためにかかる時間は、個々のファイルを別々に分離して記録する場合に比べて、非常に大きくなる。本発明は、この問題を解決する。

【0585】以上のように、AVストリームの付属情報として、前記AVストリームのClipInfoと、前記AVストリームのSTCの不連続点情報をストアするSTCInfoと、前記AVストリームのプログラム内容の不連続点情報をストアするProgramInfoと、前記AVストリーム上の時間情報とアドレス情報を関連づける情報をストアするためのCPIと、前記AVストリーム上の特徴的な画像のピクチャのタイプ（例えば番組の頭出し点）とそのピクチャのAVストリームの中のアドレスをストアするためのClipMarkを、Clip Information Fileとしてファイル化して記録媒体100に記録することにより、AVストリームの再生に必要なストリームの再生に必要なストリームの符号化情報を適切に管理することが可能である。

【0586】このClip Information file情報により、ユーザが、記録媒体100に記録されているAVストリームの中から興味のあるシーン、例えば番組の頭出し点など、をサーチすることができ、ユーザのランダムアクセスや特殊再生の指示に対して、記録媒体100からのAVストリームの読み出し位置の決定が容易になり、またストリームの復号開始を速やかに行うことができる。

【0587】上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。この場合、例えば、記録再生装置1は、図151に示されるようなパーソナルコンピュータにより構成される。

【0588】図151において、CPU（Central Processing Unit）201は、ROM（Read Only Memory）202に記憶されているプログラム、または記憶部208からRAM（Random Access Memory）203にロードされたプログラムに従って各種の処理を実行する。RAM203にはまた、CPU201が各種の処理を実行する上において必要なデータなども適宜記憶される。

【0589】CPU201、ROM202、およびRAM203は、バス204を介して相互に接続されている。このバス204にはまた、入出力インタフェース205も接続されている。

【0590】入出力インタフェース205には、キーボ

ード、マウスなどよりなる入力部206、CRT、LCDなどよりなるディスプレイ、並びにスピーカなどよりなる出力部207、ハードディスクなどより構成される記憶部208、モデム、ターミナルアダプタなどより構成される通信部209が接続されている。通信部209は、ネットワークを介しての通信処理を行う。

【0591】入出力インタフェース205にはまた、必要に応じてドライブ210が接続され、磁気ディスク221、光ディスク222、光磁気ディスク223、或いは半導体メモリ224などが適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部208にインストールされる。

【0592】上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させる場合にもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、記録媒体からインストールされる。

【0593】この記録媒体は、図151に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク221（フロッピーディスクを含む）、光ディスク222（CD-ROM（Compact Disk-Read Only Memory）、DVD（Digital Versatile Disk）を含む）、光磁気ディスク223（MD（Mini-Disk）を含む）、若しくは半導体メモリ224などよりなるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記憶されているROM202や記憶部208が含まれるハードディスクなどで構成される。

【0594】なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0595】また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【0596】

【発明の効果】以上の如く、本発明の第1の情報処理装置および方法、記録媒体、並びにプログラムによれば、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報が記録される。

【0597】本発明の第2の情報処理装置および方法、記録媒体、並びにプログラムによれば、Clip情報とし

て、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報が再生される。

【0598】従って、いずれの場合においても、AVストリームの読み出し位置の決定や復号処理を速やかに行うことができ、特に、所定のマークを迅速に検索することができる。

【図面の簡単な説明】

【図1】本発明を適用した記録再生装置の一実施の形態の構成を示す図である。

【図2】記録再生装置1により記録媒体に記録されるデータのフォーマットについて説明する図である。

【図3】Real PlaylistとVirtual Playlistについて説明する図である。

【図4】Real Playlistの作成について説明する図である。

【図5】Real Playlistの削除について説明する図である。

【図6】アセンブル編集について説明する図である。

【図7】Virtual Playlistにサブパスを設ける場合について説明する図である。

【図8】Playlistの再生順序の変更について説明する図である。

【図9】Playlist上のマークとClip上のマークについて説明する図である。

【図10】メニューサムネイルについて説明する図である。

【図11】Playlistに付加されるマークについて説明する図である。

【図12】クリップに付加されるマークについて説明する図である。

【図13】Playlist、Clip、サムネイルファイルの関係について説明する図である。

【図14】ディレクトリ構造について説明する図である。

【図15】info.dvrのシンタクスを示す図である。

【図16】DVR volumeのシンタクスを示す図である。

【図17】Resume volumeのシンタクスを示す図である。

【図18】UIAppInfovolumeのシンタクスを示す図である。

【図19】Character set valueのテーブルを示す図である。

【図20】TableOfPlaylistのシンタクスを示す図である。

【図21】TableOfPlaylistの他のシンタクスを示す図である。

【図22】MakersPrivateDataのシンタクスを示す図である。

【図23】xxxxx.rplsとyyyyy.vplsのシンタクスを示す

図である。

【図24】Playlistについて説明する図である。

【図25】Playlistのシンタクスを示す図である。

【図26】Playlist\_typeのテーブルを示す図である。

【図27】UIAppinfoPlaylistのシンタクスを示す図である。

【図28】図27に示したUIAppinfoPlaylistのシンタクス内のフラグについて説明する図である。

【図29】PlayItemについて説明する図である。

【図30】PlayItemについて説明する図である。

【図31】PlayItemについて説明する図である。

【図32】PlayItemのシンタクスを示す図である。

【図33】IN\_timeについて説明する図である。

【図34】OUT\_timeについて説明する図である。

【図35】Connection\_Conditionのテーブルを示す図である。

【図36】Connection\_Conditionについて説明する図である。

【図37】BridgeSequenceInfoを説明する図である。

【図38】BridgeSequenceInfoのシンタクスを示す図である。

【図39】SubPlayItemについて説明する図である。

【図40】SubPlayItemのシンタクスを示す図である。

【図41】SubPath\_typeのテーブルを示す図である。

【図42】PlaylistMarkのシンタクスを示す図である。

【図43】Mark\_typeのテーブルを示す図である。

【図44】Mark\_time\_stampを説明する図である。

【図45】zzzzz.clipのシンタクスを示す図である。

【図46】ClipInfoのシンタクスを示す図である。

【図47】Clip\_stream\_typeのテーブルを示す図である。

【図48】offset\_SPNについて説明する図である。

【図49】offset\_SPNについて説明する図である。

【図50】STC区間について説明する図である。

【図51】STC\_Infoについて説明する図である。

【図52】STC\_Infoのシンタクスを示す図である。

【図53】ProgramInfoを説明する図である。

【図54】ProgramInfoのシンタクスを示す図である。

【図55】VideoCondensingInfoのシンタクスを示す図である。

【図56】Video\_formatのテーブルを示す図である。

【図57】frame\_rateのテーブルを示す図である。

【図58】display\_aspect\_ratioのテーブルを示す図である。

【図59】AudioCondensingInfoのシンタクスを示す図である。

【図60】audio\_codingのテーブルを示す図である。

【図61】audio\_component\_typeのテーブルを示す図である。

【図62】sampling\_frequencyのテーブルを示す図であ

る。

【図63】CPIについて説明する図である。

【図64】CPIについて説明する図である。

【図65】CPIのシンタクスを示す図である。

【図66】CPI\_typeのテーブルを示す図である。

【図67】ビデオEP\_mapについて説明する図である。

【図68】EP\_mapについて説明する図である。

【図69】EP\_mapについて説明する図である。

【図70】EP\_mapのシンタクスを示す図である。

【図71】EP\_type valuesのテーブルを示す図である。 10

【図72】EP\_map\_for\_one\_stream\_PIDのシンタクスを示す図である。

【図73】TU\_mapについて説明する図である。

【図74】TU\_mapのシンタクスを示す図である。

【図75】ClipMarkのシンタクスを示す図である。

【図76】mark\_typeのテーブルを示す図である。

【図77】mark\_type\_stampのテーブルを示す図である。

【図78】ClipMarkのシンタクスの他の例を示す図である。

【図79】Mark\_typeのテーブルの他の例を示す図である。

【図80】mark\_entry()とrepresentative\_picture\_entry()の例を示す図である。

【図81】mark\_entry()とrepresentative\_picture\_entry()のシンタクスを示す図である。

【図82】mark\_entry()とrepresentative\_picture\_entry()のシンタクスの他の例を示す図である。

【図83】RSPN\_ref\_EP\_startとoffset\_num\_picturesの関係を説明する図である。

【図84】mark\_entry()とrepresentative\_picture\_entry()のシンタクスの他の例を示す図である。

【図85】ClipMarkとEP\_mapの関係を説明する図である。

【図86】menu.thmbとmark.thmbのシンタクスを示す図である。

【図87】Thumbnailのシンタクスを示す図である。

【図88】thumbnail\_picture\_formatのテーブルを示す図である。

【図89】tn\_blockについて説明する図である。

【図90】DVR MPEG2のトランスポートストリームの構造について説明する図である。

【図91】DVR MPEG2のトランスポートストリームのレコーダモデルを示す図である。

【図92】DVR MPEG2のトランスポートストリームのプレーヤモデルを示す図である。

【図93】source packetのシンタクスを示す図である。

【図94】TP\_extra\_headerのシンタクスを示す図である。

【図95】copy permission indicatorのテーブルを示す図である。

【図96】シームレス接続について説明する図である。

【図97】シームレス接続について説明する図である。

【図98】シームレス接続について説明する図である。

【図99】シームレス接続について説明する図である。

【図100】シームレス接続について説明する図である。

【図101】オーディオのオーバーラップについて説明する図である。

【図102】BridgeSequenceを用いたシームレス接続について説明する図である。

【図103】BridgeSequenceを用いないシームレス接続について説明する図である。

【図104】DVR STDモデルを示す図である。

【図105】復号、表示のタイミングチャートを示す図である。

【図106】図81のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

20 【図107】図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図81のシンタクスの場合における再生の動作を説明する図である。

【図108】EP\_mapの例を示す図である。

【図109】ClipMarkの例を示す図である。

【図110】図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図81のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

30 【図111】図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図81のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

【図112】図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図82のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

【図113】図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図82のシンタクスの場合における再生を説明する図である。

40 【図114】EP\_mapの例を示す図である。

【図115】ClipMarkの例を示す図である。

【図116】図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図82のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

【図117】図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図82のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

50 【図118】図75または図78のClipMarkのmark\_ent

ry()/representative\_picture\_entry()が図84のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

【図119】図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図84のシンタクスの場合における再生を説明する図である。

【図120】EP\_mapの例を示す図である。

【図121】ClipMarkの例を示す図である。

【図122】図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図84のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

【図123】図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図84のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

【図124】アプリケーションフォーマットを示す図である。

【図125】PlayList上のマークとClip上のマークを説明する図である。

【図126】ClipMarkのシンタクスの他の例を示す図である。

【図127】ClipMarkのシンタクスのさらに他の例を示す図である。

【図128】ClipInfo()のシンタクスの別の例を示す図である。

【図129】ProgramInfo()のシンタクスの別の例を示す図である。

【図130】StreamCodingInfo()のシンタクスを示す図である。

【図131】stream\_coding\_typeを説明する図である。

【図132】EP-fineとEP-coarseの関係を説明する図である。

【図133】PTS\_EP\_fineとPTS\_EP\_coarseのフォーマットを説明する図である。

【図134】RSPN\_EP\_fineとRSPN\_EP\_coarseのフォーマットを説明する図である。

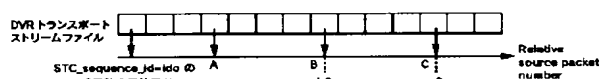
【図135】EP-coarseのエントリーとEP-fineのエントリーを説明する図である。

【図136】EP\_mapのシンタクスの別の例を示す図である。

【図137】EP\_stream\_type valuesを説明する図である。

【図138】図136のEP\_mapのEP\_map\_for\_one\_strea\*

【図107】



\* m\_PIDのシンタクスを示す図である。

【図139】EP\_video\_typeの値の意味を説明する図である。

【図140】Clip AVストリームファイルおよびClip Informationファイルの作成処理を説明するフローチャートである。

【図141】STC\_Infoの作成の動作例を説明するフローチャートである。

【図142】ProgramInfoの作成の動作例を説明するフローチャートである。

【図143】EP\_mapの作成の動作例を説明するフローチャートである。

【図144】アナログAV信号をエンコードして記録する場合における、図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図81に示すシンタクスであるときのClipMarkの作成方法を説明するフローチャートである。

【図145】デジタルインタフェースから入力されたトランスポートストリームを記録する場合における、図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図81に示すシンタクスであるときのClipMarkの作成方法を説明するフローチャートである。

【図146】EP\_mapを使う特殊再生を説明する図である。

【図147】EP\_mapを使用した1ピクチャサーチのためのプレーヤモデルを説明する図である。

【図148】ミニマイズのオペレーションの例を示す図である。

【図149】ミニマイズのときにIN\_timeの前の不要なストリームデータを消去する例を示す図である。

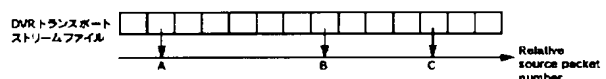
【図150】ミニマイズのときにOUT\_timeの後ろの不要なストリームデータを消去する例を説明する図である。

【図151】媒体を説明する図である。

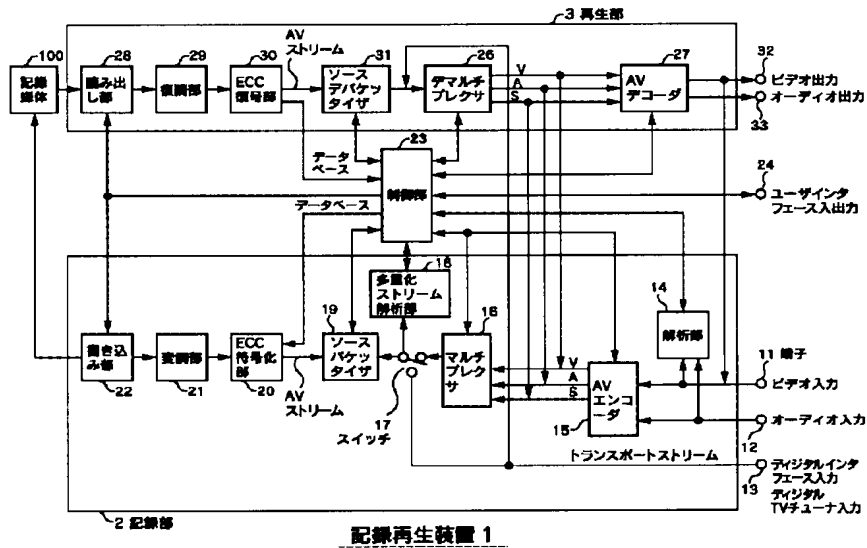
【符号の説明】

1 記録再生装置, 11乃至13 端子, 14 解析部, 15 AVエンコーダ, 16 マルチプレクサ, 17 スイッチ, 18 多重化ストリーム解析部, 19 ソースパケットタイザ, 20 ECC符号化部, 21 変調部, 22 書き込み部, 23 制御部, 24 ユーザインタフェース, 26 デマルチプレクサ, 27 AVデコーダ, 28 読み出し部, 29 復調部, 30 ECC復号部, 31 ソースパケットタイザ, 32, 33 端子

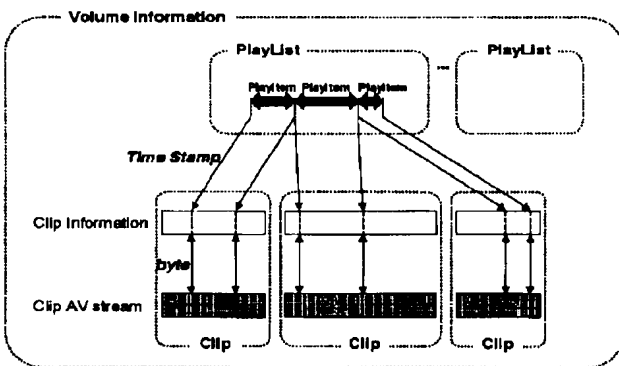
【図113】



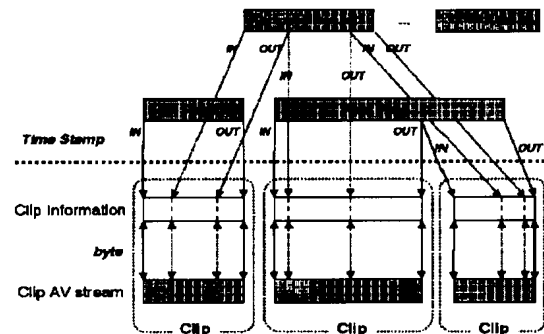
【図1】



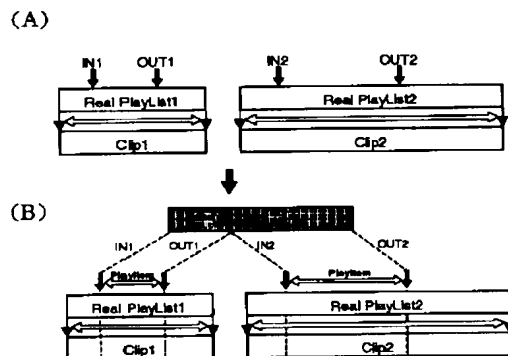
【図2】



【図3】

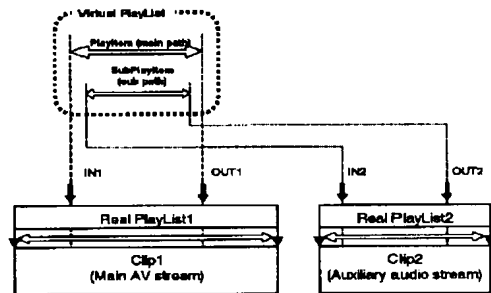


【図6】



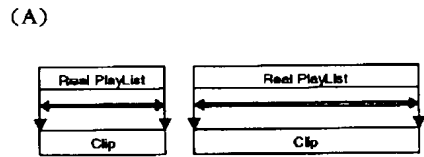
アセンブル編集の例

【図7】

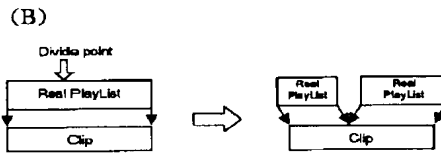


Virtual Playlist へのオーディオのアフレコの例

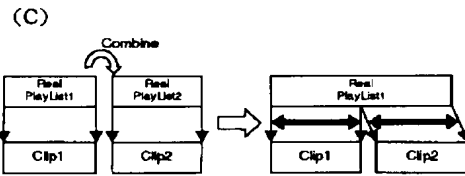
【図4】



Real PlayList のクリエイトの例

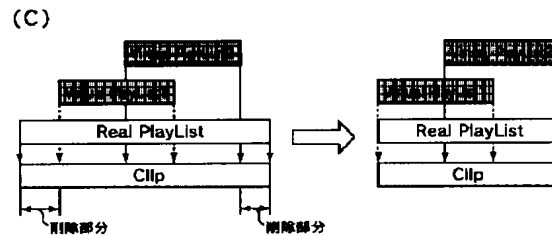
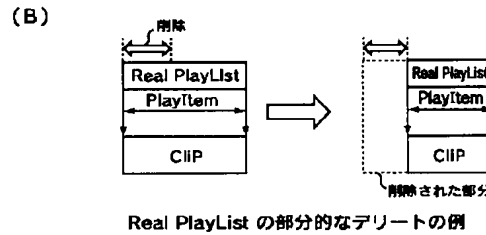
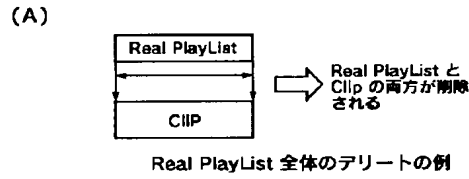


Real PlayList のディバイドの例



Real PlayList のコンバインの例

【図5】



【図108】

EP_map	
RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

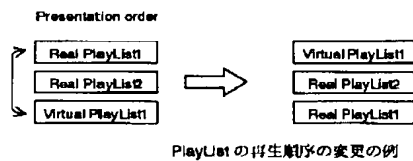
【図114】

EP_map	
RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

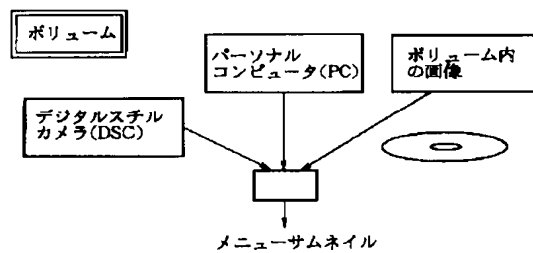
【図120】

EP_map	
RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

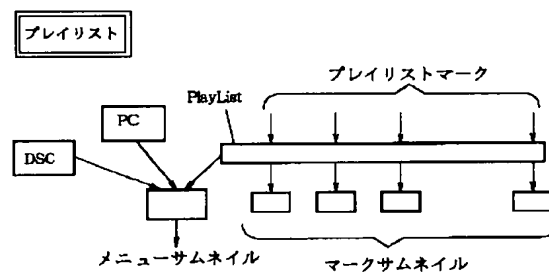
【図8】



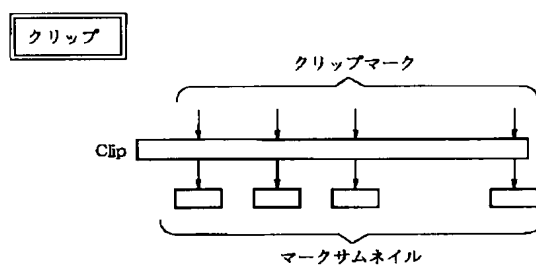
【図10】



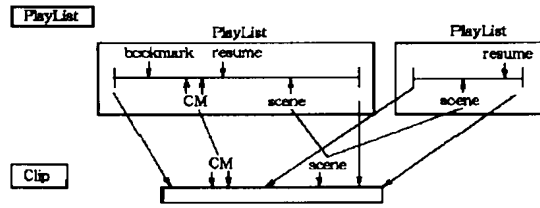
【図11】



【図12】

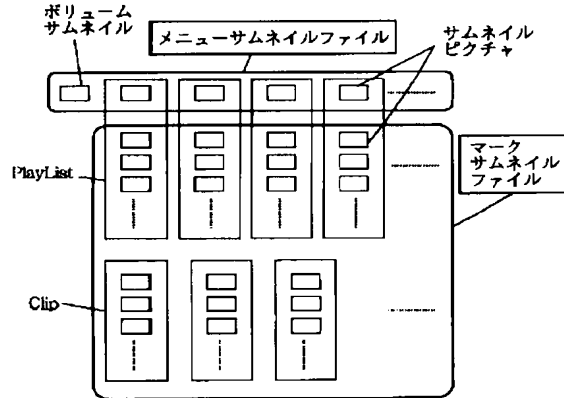


【図9】



Playlist 上のマークと Clip 上のマーク

【図13】

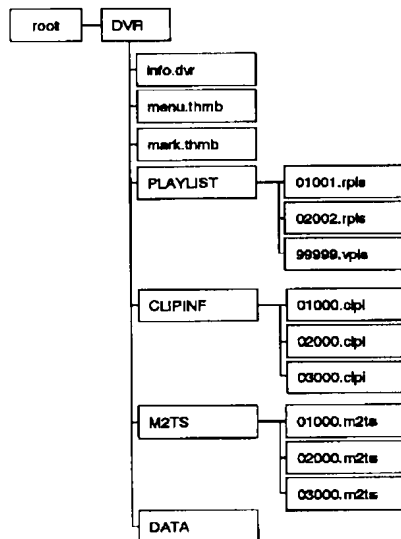


【図16】

Syntax	No. of bits	of Mnemonics
DVRVolume() {		
version_number	8*4	bslbf
length	32	umsl
ResumeVolume()		
UAppInfoVolume()		
}		

DVR Volume のシンタクス

【図14】



【図15】

Syntax	No. of bits	of Mnemonics
Info.dvr {		
TableOfPlayLists_start_address	32	umsl
MakersPrivateData_start_address	32	umsl
reserved	192	bslbf
DVRVolume()		
for(i=0; i<N1; i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for(i=0; i<N2; i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

info.dvr のシンタクス

【図19】

Value	Character coding
0x00	Reserved
0x01	ISO/IEC 846 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xFF	Reserved

Character set value



【図17】

Syntax	No. of bits	of Mnemonics
ResumeVolume() {		
reserved	15	bslbf
valid_flag	1	bslbf
resume Playlist name	8*10	bslbf
}		

ResumeVolume のシンタクス

【図18】

Syntax	No. of bits	of Mnemonics
UIAppInfoVolume() {		
character_set	8	bslbf
name_length	8	uimabf
Volume_name	8*256	bslbf
reserved	15	bslbf
Volume_protect_flag	1	bslbf
PIN	8*4	bslbf
ref_thumbnail_index	16	uimabf
reserved_for_future_use	256	bslbf
}		

UIAppInfoVolume のシンタクス

【図20】

Syntax	No. of bits	of Mnemonics
TableOfPlayLists() {		
version_number	8*4	bslbf
length	32	uimabf
number_of_PlayLists	16	uimabf
for (i=0; i<number_of_PlayLists; i++) {		
Playlist file name	8*10	bslbf
}		
}		

TableOfPlayLists のシンタクス

【図21】

## ■ TableOfPlayLists - シンタクス (4.2.3.2 の別案)

Syntax	No. of bits	of Mnemonics
TableOfPlayLists() {		
version_number	8*4	bslbf
length	32	uimabf
number_of_PlayLists	16	uimabf
for (i=0; i<number_of_PlayLists; i++) {		
Playlist file name	8*10	bslbf
UIAppInfoPlaylist()		
}		
}		

TableOfPlayLists の別シンタクス

【図23】

Syntax	No. of bits	of Mnemonics
xxxxx.rpls / yyyyy.vpls {		
PlaylistMark_Start_address	32	uimabf
MakersPrivateData_Start_address	32	uimabf
reserved	192	bslbf
Playlist()		
for(i=0; i<N1; i++){		
padding_word	16	bslbf
}		
PlaylistMark()		
for(i=0; i<N2; i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

xxxxx.rpls と yyyyy.vpls のシンタクス

【図26】

Playlist_type	Meaning
0	AV記録のための Playlist この Playlist に参照されるすべての Clip は、一つ以上のビデオストリームを含まなければならない。
1	オーディオ記録のための Playlist この Playlist に参照されるすべての Clip は、一つ以上のオーディオストリームを含まなければならない、そしてビデオストリームを含んではならない。
2 - 255	reserved

Playlist\_type

【図28】

(A)

write_protect_flag	Meaning
0b	その Playlist を自由に消去しても良い。
1b	write_protect_flag を除いてその Playlist の内容は、消去および変更されるべきではない。

write\_protect\_flag

(B)

is_played_flag	Meaning
0b	その Playlist は、記録されてから一度も再生されたことがない。
1b	Playlist は、記録されてから一度は再生された。

is\_played\_flag

(C)

archive	Meaning
00b	何も情報が定義されていない。
01b	オリジナル
10b	コピー
11b	reserved

archive

【図22】

Syntax	No. bits	of	Mnemonics
MakersPrivateData() {			
version number	8*4		bslbf
length	32		uimbsf
if (length != 0) {			
mpd_blocks_start_address	32		uimbsf
number of maker entries	16		uimbsf
mpd_block size	16		uimbsf
number of mpd blocks	16		uimbsf
reserved	16		bslbf
for (i=0; i<number of maker entries; i++) {			
maker ID	16		uimbsf
maker model code	16		uimbsf
start mpd block number	16		uimbsf
reserved	16		bslbf
mpd length	32		uimbsf
}			
stuffing bytes	8*2*L1		bslbf
for (i=0; i<number of mpd blocks; i++) {			
mpd_block	mpd_block_size*1024*8		
}			
}			

MakersPrivateData のシンタックス

【図25】

Syntax	No. bits	of	Mnemonics
Playlist() {			
version number	8*4		bslbf
length	32		uimbsf
Playlist_type	8		uimbsf
CPI_type	1		bslbf
reserved	7		bslbf
UIAppinfoPlaylist()			
number of PlayItems // main path	16		uimbsf
if (<Virtual Playlist>) {			
number of SubPlayItems // sub path	16		uimbsf
} else {			
reserved	16		bslbf
}			
for (PlayItem_id=0; PlayItem_id<number of PlayItems; PlayItem_id++) {			
PlayItem() // main path			
}			
if (<Virtual Playlist>) {			
if (CPI_type==0 && Playlist_type==0) {			
for (i=0; i<number of SubPlayItems; i++) {			
SubPlayItem() // sub path			
}			
}			
}			

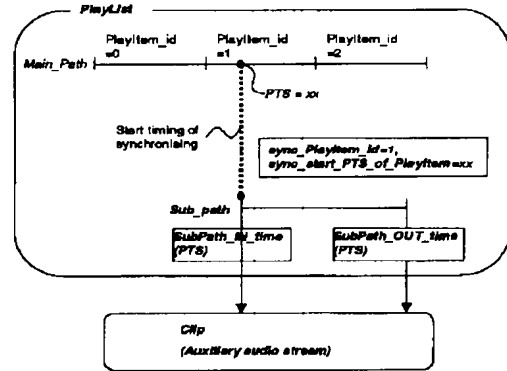
Playlist のシンタックス

【図33】

CPI_type in the Playlist()	Semantics of IN_time
EP_map type	IN_time は、PlayItem の中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。
TU_map type	IN_time は、TU_map_time_axis上の時刻でなければならない。かつ、IN_time は、time_unitの精度に丸めて表さなければならない。IN_time は、次に示す等式により計算される。
	$IN\_time = TU\_start\_time \% 2^{32}$

IN\_time

【図39】



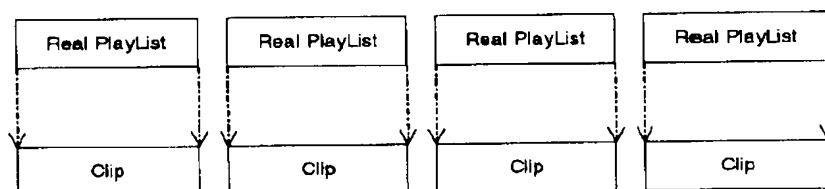
【図41】

SubPath_type	Meaning
0x00	Auxiliary audio stream path
0x01 ~ 0xff	reserved

SubPath\_type

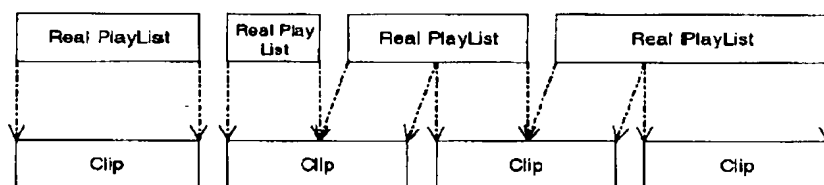
【図24】

(A)



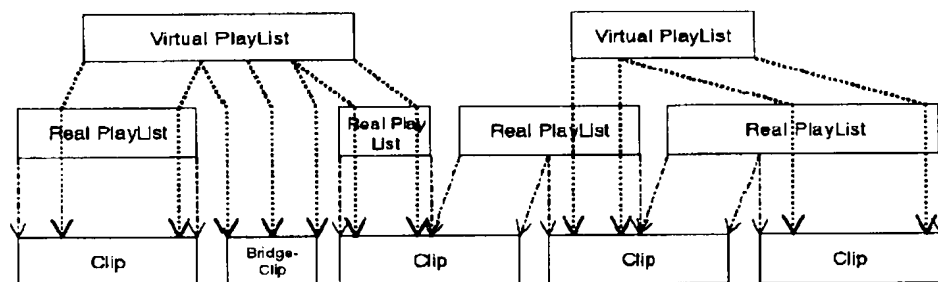
初めて AV ストリームが Clip として記録された時の Real PlayList の例

(B)



編集後の Real PlayList の例

(C)



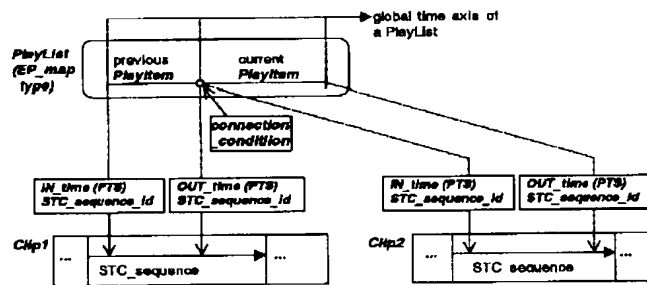
Virtual PlayList の例

【図27】

Syntax	No. of bits	of	Mnemonics
UIAppInfoPlayList2()			
character set	8		bslbf
name length	8		ulmsbf
PlayList name	8*256		bslbf
reserved	8		bslbf
record time and date	4*14		bslbf
reserved	8		bslbf
duration	4*8		bslbf
valid period	4*8		bslbf
maker id	16		ulmsbf
maker code	16		ulmsbf
reserved	11		bslbf
playback control flag	1		bslbf
write protect flag	1		bslbf
is played flag	1		bslbf
archive	2		bslbf
ref thumbnail index	16		ulmsbf
reserved for future use	256		bslbf

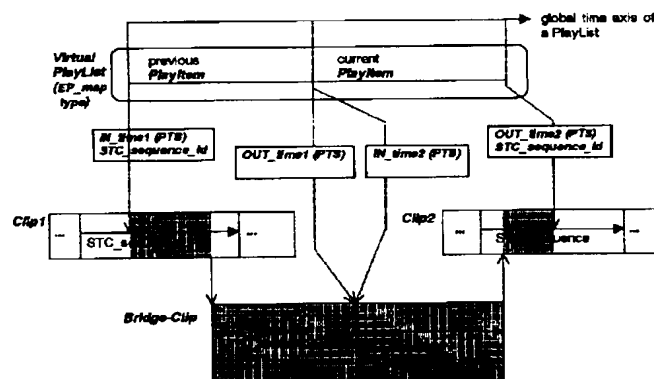
UIAppInfoPlayList のシンタクス

【図29】



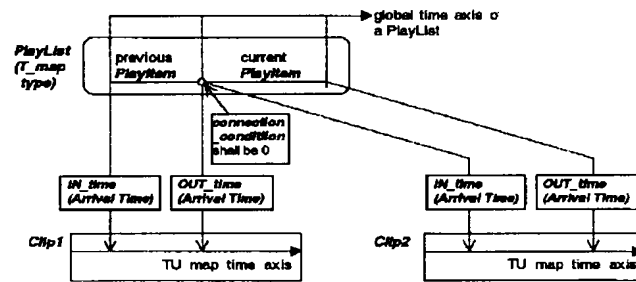
PlayList が EP\_map type であり、かつ PlayItem が BridgeSequence を持たない時の例

【図30】



PlayList が EP\_map type であり、かつ PlayItem が BridgeSequence を持つ時の例

【図31】



PlayList が TU\_map type である時の例

【図32】

Syntax	No. of bits	Mnemonics
PlayItem() {		
Clip information file name	8*10	bslbf
reserved	24	bslbf
STC sequence Id	8	uimsbf
IN time	32	uimsbf
OUT time	32	uimsbf
reserved	14	bslbf
connection condition	2	bslbf
if (<Virtual PlayList> {		
if (connection_condition=="10") {		
BridgeSequenceInfo()		
}		
}		
}		

PlayItem のシンタクス

【図34】

Clip type in the PlayList()	Semantics of OUT_time
EP_map type	<p>OUT_time は、次に示す等式によって計算される Presentation_end_TS の値の上位 32 ビットを示さなければならない。</p> $\text{Presentation\_end\_TS} = \text{PTS\_out} + \text{AU\_duration}$ <p>ここで、</p> <p>PTS_out は、PlayItem の中で最後のプレゼンテーションユニットに対応する 33 ビット長の PTS である。</p> <p>AU_duration は、最後のプレゼンテーションユニットの 90kHz 単位の表示期間である。</p>
TU_map type	<p>OUT_time は、TU_map_time_axis 上の時刻でなければならない。かつ、OUT_time は、time_unit の精度に丸めて表さなければならない。OUT_time は、次に示す等式により計算される。</p> $\text{OUT\_time} = \text{TU\_start\_time} \% 2^{32}$

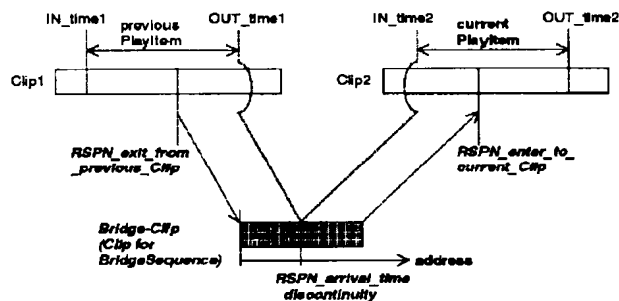
OUT\_time

【図35】

connection condition	meaning
00	<ul style="list-style-type: none"> <li>先行する PlayItem と現在の PlayItem の接続は、シームレス再生の保証がなされていない。</li> <li>PlayList の CPI_type が TU_map type である場合、connection_condition は、この値をセットされなければならない。</li> </ul>
01	<ul style="list-style-type: none"> <li>この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。</li> <li>先行する PlayItem と現在の PlayItem は、システムタイムベース (STC ベース) の不連続点があるために分割されていることを表す。</li> </ul>
10	<ul style="list-style-type: none"> <li>この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。</li> <li>この状態は、Virtual PlayList に対してだけ許される。</li> <li>先行する PlayItem と現在の PlayItem との接続は、シームレス再生の保証がなされている。</li> <li>先行する PlayItem と現在の PlayItem は、BridgeSequence を使用して接続されており、DVR MPEG-2 トランスポートストリームは、後述する DVR-STD に従っていないなければならない。</li> </ul>
11	<ul style="list-style-type: none"> <li>この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。</li> <li>先行する PlayItem と現在の PlayItem は、シームレス再生の保証がなされている。</li> <li>先行する PlayItem と現在の PlayItem は、BridgeSequence を使用しないで接続されており、DVR MPEG-2 トランスポートストリームは、後述する DVR-STD に従っていないなければならない。</li> </ul>

connection\_condition

【図37】

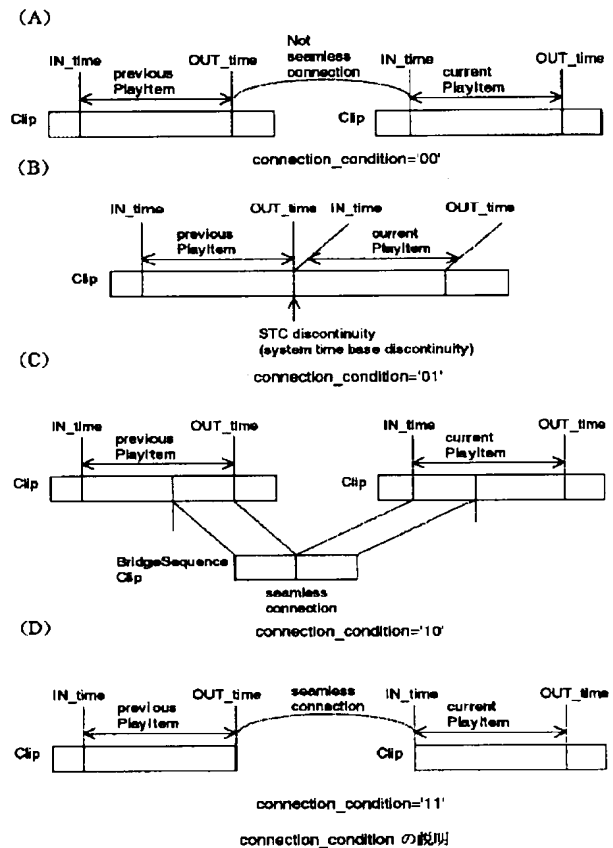


【図47】

Clip stream type	meaning
0	Clip AV ストリーム
1	Bridge-Clip AV ストリーム
2 - 255	Reserved

Clip\_stream\_type

【図36】



connection\_conditionの説明

【図44】

CPI_type in the PlayList()	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	mark_time_stamp は、TU_map_time_axis 上の時刻でなければならない。かつ、mark_time_stamp は、time unit の精度に丸めて表さなければならない。mark_time_stamp は、次に示す等式により計算される。 $\text{mark\_time\_stamp} = \text{TU\_start\_time} \% 2^{32}$

mark\_time\_stamp

【図38】

Syntax	No. of bits	of Mnemonics
BridgeSequenceInfo {		
Bridge Clip information file name	8*10	bslbf
RSPN exit from previous Clip	32	uimabf
RSPN enter to current Clip	32	uimabf
}		

BridgeSequenceInfo のシンタクス

【図40】

Syntax	No. of bits	of Mnemonics
SubPlayItem() {		
Clip information file name	8*10	bslbf
SubPath type	8	bslbf
sync PlayItem id	8	uimabf
sync start PTS of PlayItem	32	uimabf
SubPath IN time	32	uimabf
SubPath OUT time	32	uimabf
}		

SubPlayItem のシンタクス

【図56】

video_format	Meaning
0	480i
1	576i
2	480p (including 840x480p format)
3	1080i
4	720p
5	1080p
6 - 254	reserved
255	No information

video\_format

【図42】

Syntax	No. of bits	of Mnemonics
PlayListMark() {		
version number	8*4	bslbf
length	32	uimabf
number of PlayList marks	16	uimabf
for(i=0; i<number of PlayList marks; i++) {		
reserved	8	bslbf
mark type	8	bslbf
mark time etemp	32	uimabf
PlayItem id	8	uimabf
reserved	24	uimabf
character set	8	bslbf
name length	8	uimabf
mark name	8*256	bslbf
ref thumbnail index	16	uimabf
}		
}		

PlayListMark のシンタクス

【図43】

Mark type	Meaning	Comments
0x00	resume-mark	再生リジュームポイント。PlayListMark()において定義される再生リジュームポイントの数は、0または1でなければならない。
0x01	book-mark	PlayListの再生エントリーポイント。このマークは、ユーザがセットすることができ、例えば、お気に入りのシーンの開始点を指定するマークに使う。
0x02	skip-mark	スキップマークポイント。このポイントからプログラムの最後まで、プレーヤはプログラムをスキップする。PlayListMark()において定義されるスキップマークポイントの数は、0または1でなければならない。
0x03 - 0x8F	reserved	
0x90 - 0xFF	reserved	Reserved for ClipMark()

mark\_type

【図45】

Syntax	No. of bits	of	Mnemonics
zzzz.cipl {			
STC_Info_Start_address	32		uintof
ProgramInfo_Start_address	32		uintof
CPI_Start_address	32		uintof
ClipMark_Start_address	32		uintof
MakersPrivateData_Start_address	32		uintof
reserved	96		bsbf
ClipInfo()			
for(i=0; i<N1; i++){			
padding_word	16		bsbf
}			
STC_Info()			
for(i=0; i<N2; i++){			
padding_word	16		bsbf
}			
ProgramInfo()			
for(i=0; i<N3; i++){			
padding_word	16		bsbf
}			
CPI()			
for(i=0; i<N4; i++){			
padding_word	16		bsbf
}			
ClipMark()			
for(i=0; i<N5; i++){			
padding_word	16		bsbf
}			
MakersPrivateData()			
}			

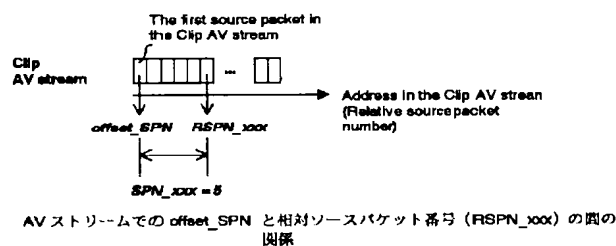
zzzz.ciplのシンタクス

【図46】

Syntax	No. of bits	of	Mnemonics
ClipInfo {			
version number	8*4		bsbf
length	32		uintof
Clip_stream_type	8		bsbf
offset_SPN	32		uintof
TS recording rate	24		uintof
reserved	8		bsbf
record time and date	4*14		bsbf
reserved	8		bsbf
duration	4*6		bsbf
reserved	7		bsbf
time controlled flag	1		bsbf
TS average rate	24		uintof
// (Clip stream type==1) // Bridge-Clip AV stream			
RSPN arrival time discontinuity	32		uintof
else			
reserved	32		bsbf
reserved_for_system_use	144		bsbf
reserved	11		bsbf
is format identifier valid	1		bsbf
is original network ID valid	1		bsbf
is transport stream ID valid	1		bsbf
is service ID valid	1		bsbf
is country code valid	1		bsbf
format identifier	32		bsbf
original network ID	16		uintof
transport stream ID	16		uintof
service ID	16		uintof
country code	24		bsbf
stream format name	16*8		bsbf
reserved_for_future_use	256		bsbf
}			

ClipInfoのシンタクス

【図49】



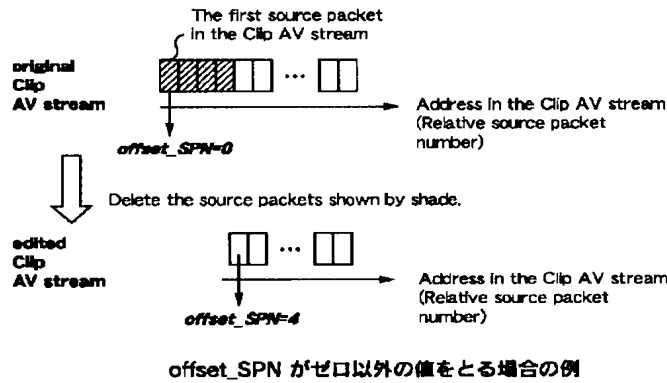
【図57】

frame_rate	Meaning
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97...)
5	30
6	50
7	60 000/1001 (59.94...)
8	60
9 - 254	reserved
255	No information

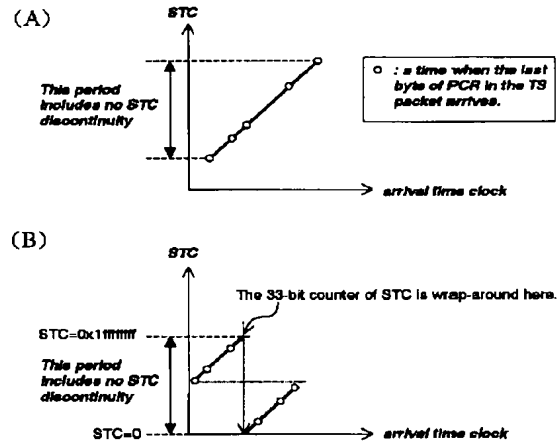
frame\_rate



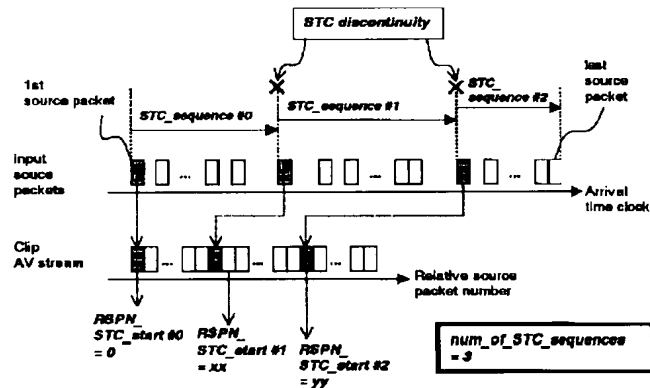
【図48】



【図50】



【図51】



--STC\_Info

【図66】

CPI type	Meaning
0	EP map type
1	TU map type

CPI\_type の意味

【図52】

Syntax	No. of bits	of	Mnemonics
STC_Info {			
version number	8*4		bslbf
length	32		uimabf
if (length != 0) {			
reserved	8		bslbf
num_of_STC_sequences	8		uimabf
for(STC_sequence_id=0;			
STC_sequence_id < num_of_STC_sequences;			
STC_sequence_id++) {			
reserved	32		bslbf
RSPN_STC_start	32		uimabf
}			
}			

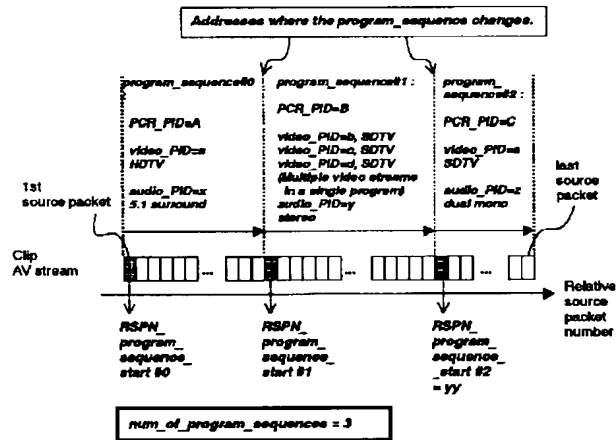
STC\_Info のシンタクス

【図61】

audio_component_type	Meaning
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-254	reserved
255	No information

audio\_component\_type

【図53】



ProgramInfo の例

【図54】

Syntax	No. of bits	Mnemonics
ProgramInfo {		
version number	8*4	balbf
length	32	uimabf
if (length != 0) {		
reserved	8	balbf
number of program sequences	8	uimabf
for (i=0; i<number of program sequences; i++){		
RSPN_program_sequence_start	32	uimabf
reserved	48	balbf
PCR_PID	16	balbf
number of videos	8	uimabf
number of audios	8	uimabf
for (k=0; k<number of videos; k++){		
video stream PID	16	balbf
VideoCodingInfo()		
}		
for (k=0; k<number of audios; k++){		
audio stream PID	16	balbf
AudioCodingInfo()		
}		
}		
}		

ProgramInfo のシンタクス

【図55】

Syntax	No. of bits	Mnemonics
VideoCodingInfo {		
video format	8	uimabf
frame rate	8	uimabf
display_aspect_ratio	8	uimabf
reserved	8	balbf
}		

VideoCodingInfo のシンタクス

【図58】

display_aspect_ratio	Meaning
0	forbidden
1	reserved
2	4:3 display aspect ratio
3	16:9 display aspect ratio
4-254	reserved
255	No information

display\_aspect\_ratio

【図62】

sampling_frequency	Meaning
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

sampling\_frequency

【図59】

Syntax	No. of bits	of Mnemonics
AudioCodingInfo()		
audio_coding	8	uint8
audio_component_type	8	uint8
sampling_frequency	8	uint8
reserved	8	bit

AudioCodingInfoのシンタクス

【図60】

audio_coding	Meaning
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No information

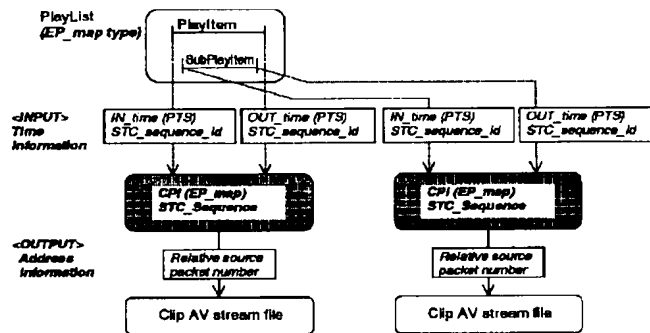
audio\_coding

【図76】

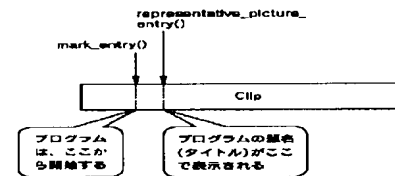
Mark type	Meaning	Comments
0x00 - 0x0F	reserved	Reserved for PlaylistMark()
0x10	Event-start mark	番組の開始ポイントを示すマーク点。
0x11	Local event-start mark	番組の中の局所的な場面を示すマーク点。
0x12	Scene-start mark	シーンチェンジポイントを示すマーク。
0x13 - 0xFF	reserved	

mark\_type

【図63】



【図80】

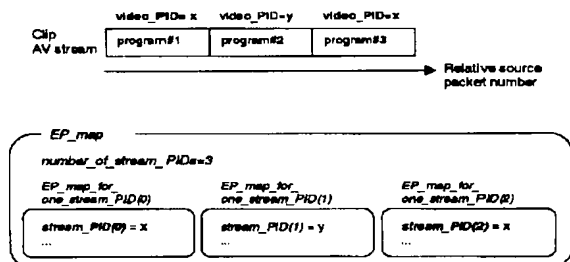


【図65】

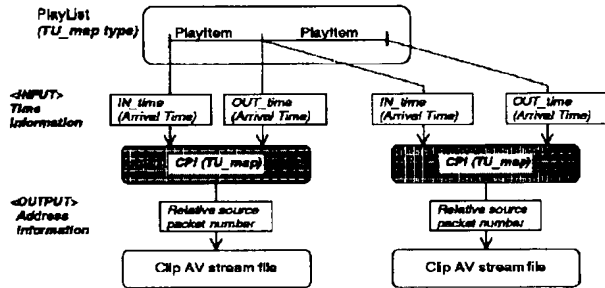
Syntax	No. of bits	of Mnemonics
CPI()		
version number	8*4	bit
length	32	uint8
reserved	15	bit
CPI_type	1	bit
if (CPI_type == 0)		
EP_map()		
else		
TU_map()		

CPIのシンタクス

【図69】



【図64】

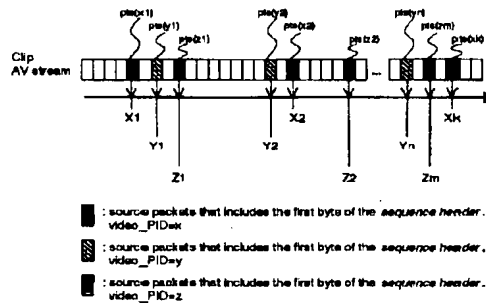


【図88】

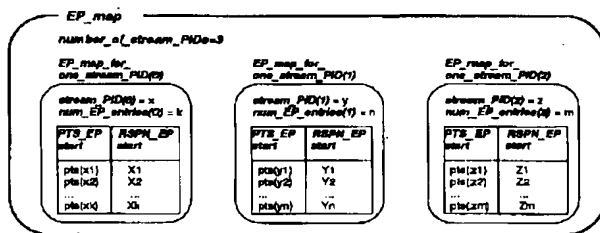
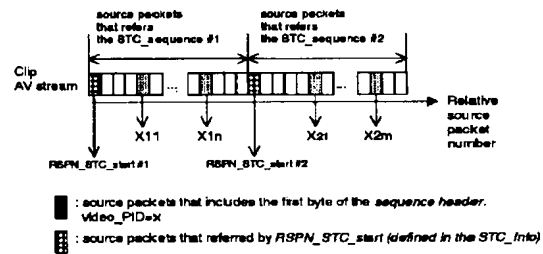
Thumbnail picture format	Meaning
0x00	MPEG-2 Video I-picture
0x01	DCF (restricted JPEG)
0x02	PNG
0x03-0xff	reserved

thumbnail\_picture\_format

【図67】



【図68】



ビデオの EP\_map の例

【図72】

Syntax	No. bits	of	Mnemonics
EP_map_for_one_stream_PID(N){			
for (i=0; i<N; i++){			
PTS_EP_start	32		uimabf
RSPN_EP_start	32		uimabf
}			

EP\_map\_for\_one\_stream\_PID のシンタクス

【図78】

Syntax	No. bits	of	Mnemonics
ClipMark0{			
version_number	8*4		ba1bf
length	32		uimabf
number_of_Clip_marks	16		uimabf
for (i=0; i<number_of_Clip_marks; i++){			
reserved	8		ba1bf
mark_type	8		ba1bf
reserved_for_MarkerID	16		ba1bf
mark_entry()			
representative_picture_entry()			
ref_thumbnail_index	16		uimabf
}			

【図70】

Syntax	No. of bits	Mnemonics
EP_map{		
reserved	12	bsbf
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0;k<number_of_stream_PIDs;k++){		
stream_PID(k)	16	bsbf
num_EP_entries(k)	32	uimsbf
EP_map_for_one_stream_PID_start_address(k)	32	uimsbf
}		
for (i=0;i<X;i++){		
padding_word	16	bsbf
}		
for (k=0;k<number_of_stream_PIDs;k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for (l=0;l<Y;l++){		
padding_word	16	bsbf
}		
}		
}		

【図71】

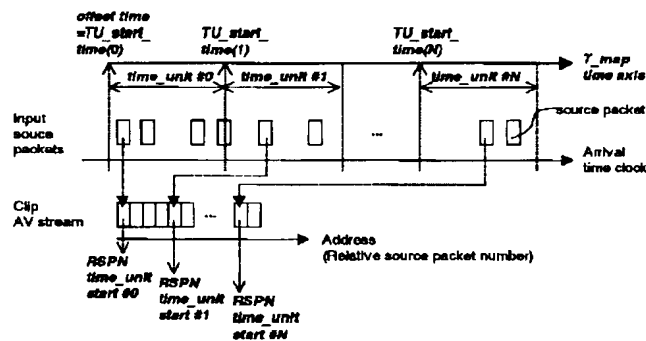
EP_type	Meaning
0	video
1	audio
2 - 15	reserved

EP\_type Values

【図81】

Syntax	No. of bits	Mnemonics
mark_entry() / representative_picture_entry() {		
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	bsbf
}		

【図73】



【図79】

Mark_type	Meaning	Comments
0x00 - 0xBF	reserved	Reserved for PlaylistMark()
0xC0	Event-start mark	番組の開始ポイントを示すマーク点
0xC1	Local event-start mark	番組の中の局所的な場面を示すマーク点
0xC2	Scene-start mark	シーン開始ポイントを示すマーク点
0xC3	Scene-end mark	シーン終了ポイントを示すマーク点
0xC4	CM-start mark	CM開始ポイントを示すマーク点
0xC5	CM-end mark	CM終了ポイントを示すマーク点
0xC6 - 0xBF	DVR フォーマットが、ClipMark を将来、拡張するのために予約されている	
0xC0 - 0xFF	メーカー独自のアプリケーションで利用するマークに割り当て可能	

【図74】

Syntax	No. of bits	Mnemonics
TU_map{		
offset_time	32	bsbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0;k<number_of_time_unit_entries;k++){		
RSPN_time_unit_start	32	uimsbf
}		

TU\_map のシンタクス

【図75】

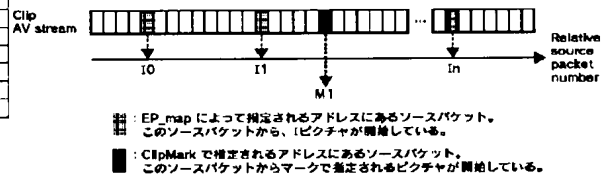
Syntax	No. of bits	of Mnemonics
ClipMark() {		
version_number	8*4	belbf
length	32	uimabf
number_of_Clip_marks	16	uimabf
for(i=0; i<number_of_Clip_marks; i++) {		
reserved	8	belbf
mark_type	8	belbf
mark_time_stamp	32	uimabf
STC_sequence_id	8	uimabf
reserved	24	belbf
character_set	8	belbf
name_length	8	uimabf
mark_name	8*256	belbf
ref_thumbnail_index	16	uimabf
}		

ClipMark のシンタクス

【図82】

Syntax	No. of bits	of Mnemonics
mark_entry()/representative_picture_entry() {		
RSPN_ref_EP_start	32	uimabf
offset_num_pictures	32	uimabf
}		

【図85】



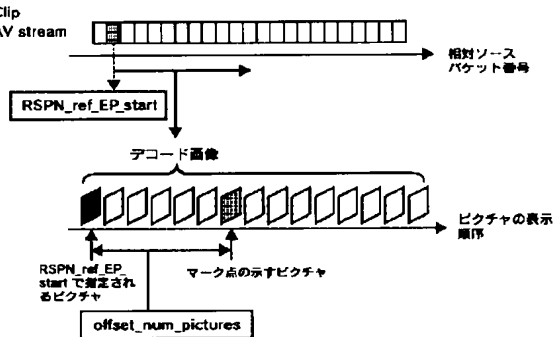
【図77】

CPI_type in the CPI()	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	mark_time_stamp は、TU_map_time_axis 上の時刻でなければならない。かつ、mark_time_stamp は、time_unit の精度に丸めて表さなければならない。mark_time_stamp は、次に示す等式により計算される。

$$\text{mark\_time\_stamp} = \text{TU\_start\_time} \% 2^{32}$$

mark\_type\_stamp

【図83】



【図84】

Syntax	No. of bits	of Mnemonics
mark_entry()/representative_picture_entry() {		
RSPN_mark_point	32	uimabf
}		

【図86】

【図95】

copy_permission indicator	meaning
00	copy free
01	no more copy
10	copy once
11	copy prohibited

copy permission indicator table

Syntax	No. of bits	of Mnemonics
menu.thmb / mark.thmb {		
reserved	256	belbf
Thumbnail()		
for(i=0; i<N1; i++)		
padding_word	16	belbf
}		

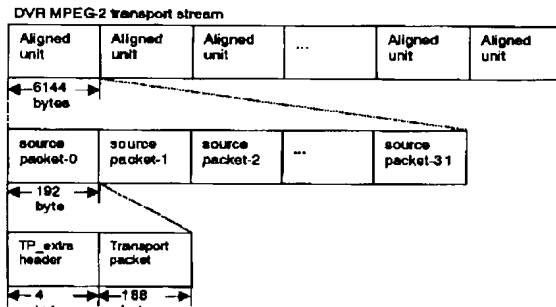
menu.thmb と mark.thmb のシンタクス

【図87】

Syntax	Bits	Mnemonics
Thumbnail() {		
version number	8*4	char
length	32	uimsbf
if (length != 0) {		
tn_block_start address	32	bits
number of thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number of tn blocks	16	uimsbf
reserved	16	bits
for(i = 0; i < number of thumbnails; i++) {		
thumbnail index	16	uimsbf
thumbnail picture format	8	bits
reserved	8	bits
picture data size	32	uimsbf
start tn block number	16	uimsbf
x picture length	16	uimsbf
y picture length	16	uimsbf
reserved	16	uimsbf
}		
stuffing bytes	8*2*L1	bits
for(k = 0; k < number of tn blocks; k++) {		
tn_block	tn_block_size*16*4*8	
}		
}		

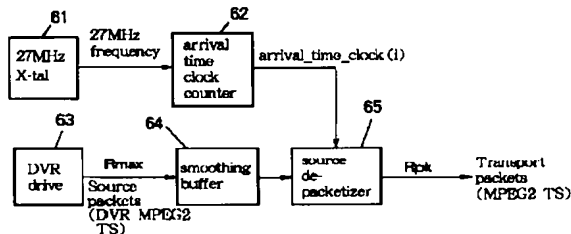
Thumbnailのシンタクス

【図90】



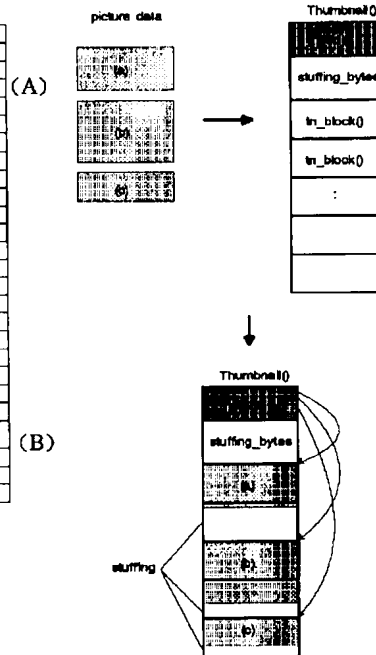
DVR MPEG-2 トランスポートストリームの構造

【図92】

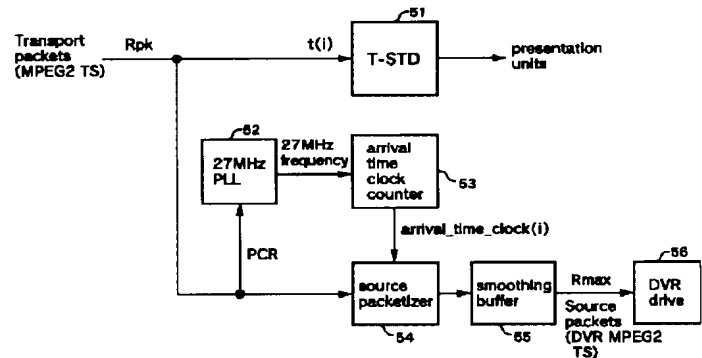


DVR MPEG-2 トランスポートストリームのプレーヤモデル

【図89】



【図91】



DVR MPEG-2 トランスポートストリームのレコーダモデル

【図93】

Syntax	No. of bits	Mnemonics
source_packet() {		
TP_extra_header()		
transport_packet()		
}		

source packet

【図75】

Syntax	No. of bits	of Mnemonics
ClipMark() {		
version_number	8*4	belbf
length	32	uimsbf
number_of_Clip_marks	16	uimsbf
for(i=0; i<number_of_Clip_marks; i++) {		
reserved	8	belbf
mark_type	8	belbf
mark_time_stamp	32	uimsbf
STC_sequence_id	8	uimsbf
reserved	24	belbf
character_set	8	belbf
name_length	8	uimsbf
mark_name	8*256	belbf
ref_thumbnail_index	16	uimsbf
}		

ClipMark のシンタクス

【図77】

CPI_type in the CPI()	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	mark_time_stamp は、TU_map_time_axis 上の時刻でなければならない。かつ、mark_time_stamp は、time_unit の精度に丸めて表さなければならない。mark_time_stamp は、次に示す等式により計算される。

$$\text{mark\_time\_stamp} = \text{TU\_start\_time} \% 2^{32}$$

mark\_type\_stamp

【図84】

Syntax	No. of bits	of Mnemonics
mark_entry()/representative_picture_entry() {		
RSPN_mark_point	32	uimsbf
}		

【図86】

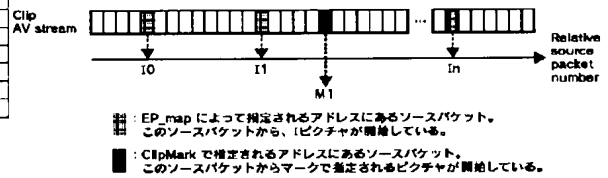
Syntax	No. of bits	of Mnemonics
menu.thmb / mark.thmb {		
reserved	256	belbf
Thumbnail() {		
for(i=0; i<N1; i++)		
padding_word	16	belbf
}		

menu.thmb と mark.thmb のシンタクス

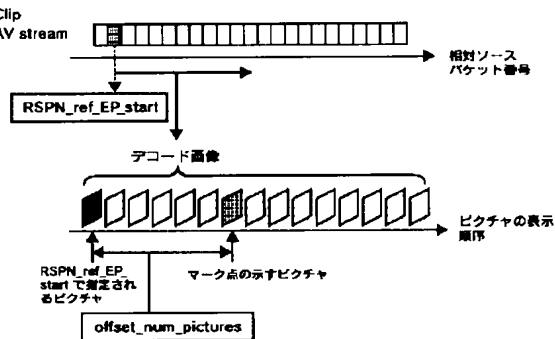
【図82】

Syntax	No. of bits	of Mnemonics
mark_entry()/representative_picture_entry() {		
RSPN_ref_EP_start	32	uimsbf
offset_num_pictures	32	uimsbf
}		

【図85】



【図83】



【図95】

copy_permission indicator	meaning
00	copy free
01	no more copy
10	copy once
11	copy prohibited

copy permission indicator table

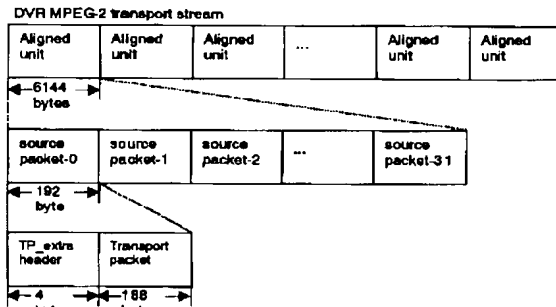


【図87】

Syntax	Bits	Mnemonics
Thumbnail() {		
version number	8*4	char
length	32	uimsbf
if (length != 0) {		
tn_block_start address	32	bits
number of thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number of tn blocks	16	uimsbf
reserved	16	bits
for(i = 0; i < number of thumbnails; i++) {		
thumbnail index	16	uimsbf
thumbnail picture format	8	bits
reserved	8	bits
picture data size	32	uimsbf
start tn block number	16	uimsbf
x picture length	16	uimsbf
y picture length	16	uimsbf
reserved	16	uimsbf
}		
stuffing bytes	8*2*L1	bits
for(k = 0; k < number of tn blocks; k++) {		
tn_block	tn_block_size*16*8	
}		
}		

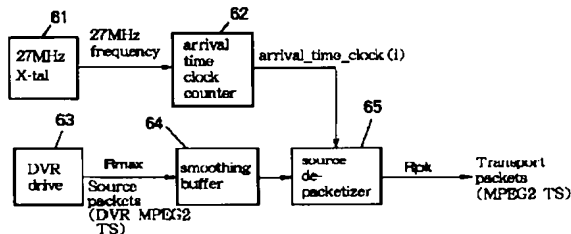
Thumbnailのシンタクス

【図90】



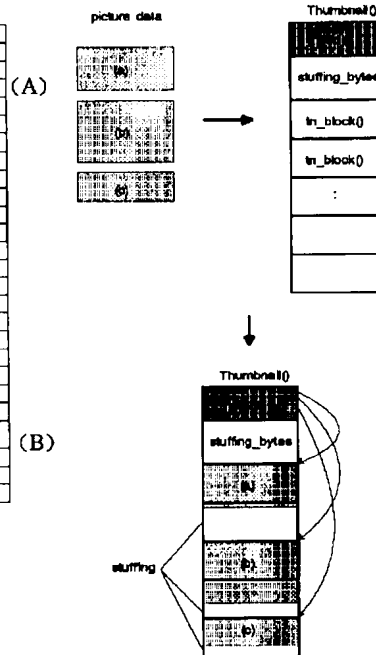
DVR MPEG-2 トランスポートストリームの構造

【図92】

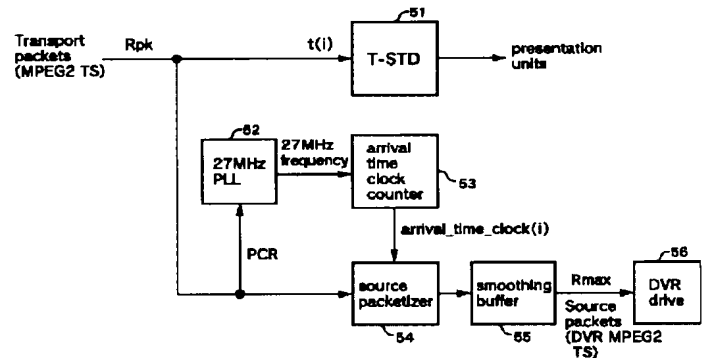


DVR MPEG-2 トランスポートストリームのプレーヤモデル

【図89】



【図91】



DVR MPEG-2 トランスポートストリームのレコーダモデル

【図93】

Syntax	No. of bits	Mnemonics
source_packet() {		
TP_extra_header()		
transport_packet()		
}		

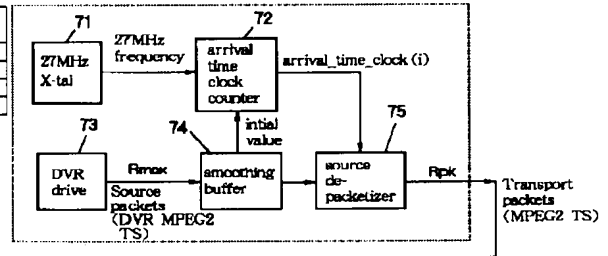
source packet

【図94】

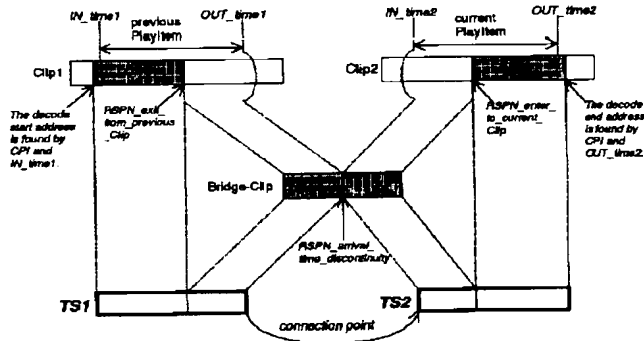
Syntax	No. of bits	Minemomics
TP_extra_header() {		
copy_permission_indicator	2	uimebf
arrival_time_stamp	30	uimebf
}		

TP\_extra\_header

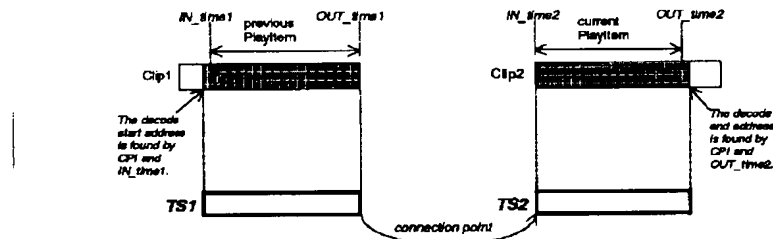
【図104】



【図96】



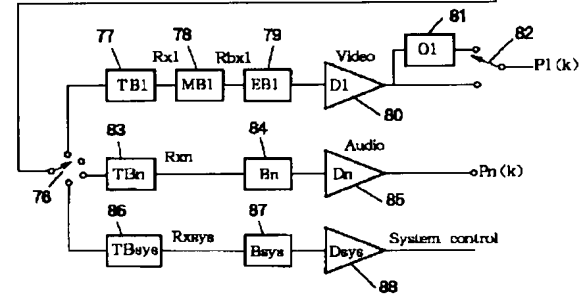
【図97】



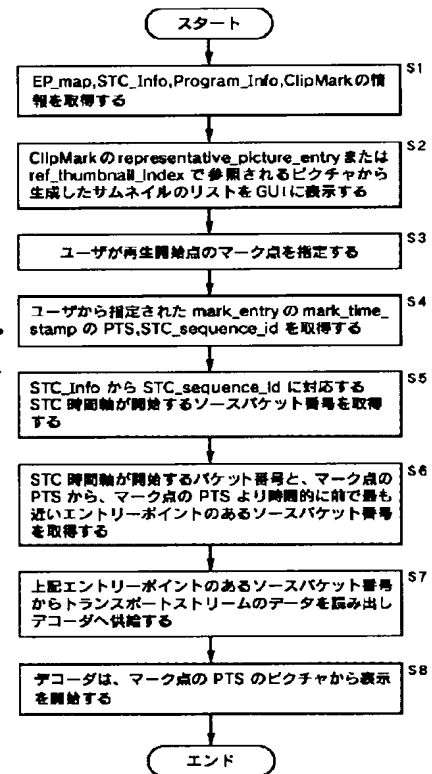
【図109】

ClipMark

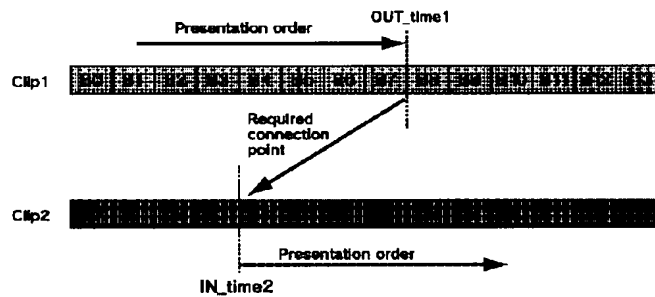
Mark_type	mark_entry		representative_picture_entry	
	Mark_Time_stamp	STC_sequence_id	Mark_Time_stamp	STC_sequence_id
...	...	...	...	...
0x92(scene start)	PTS(a1)	Id0	PTS(a2)	Id0
0x94(CMstart)	PTS(b0)	Id0	PTS(b0)	Id0
0x95(CMend)	PTS(c0)	Id0	PTS(c0)	Id0
...	...	...	...	...



【図106】

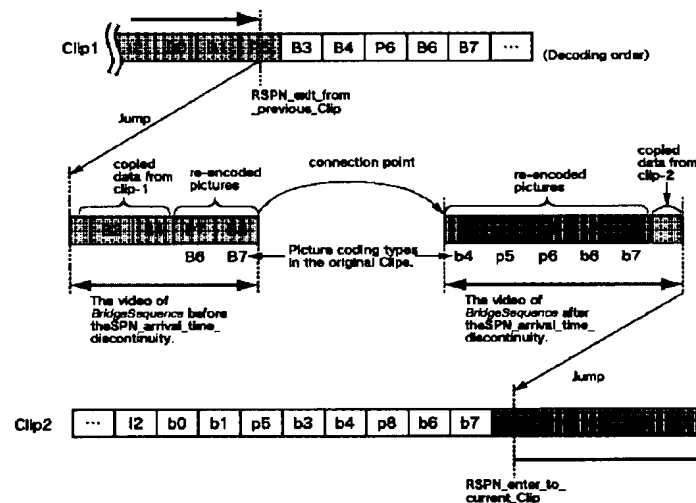


【図98】



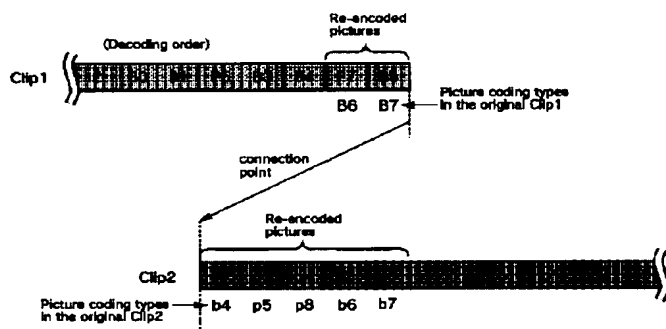
ピクチャの表示順序で示すシームレス接続の例

【図99】



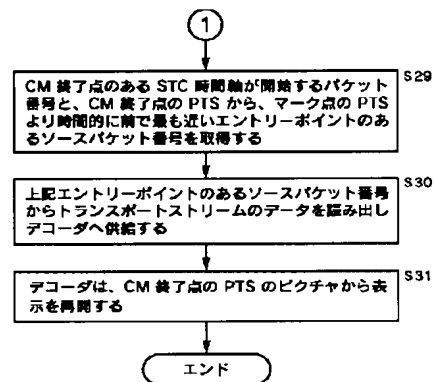
BridgeSequence を使用してシームレス接続を実現する例1

【図100】

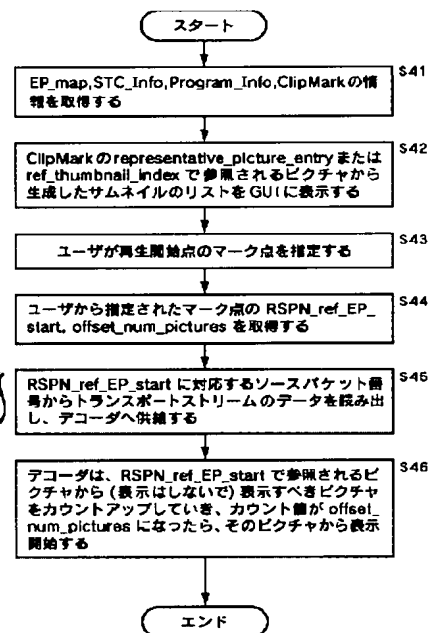


BridgeSequence を使用しないでシームレス接続を実現する例2

【図111】



【図112】

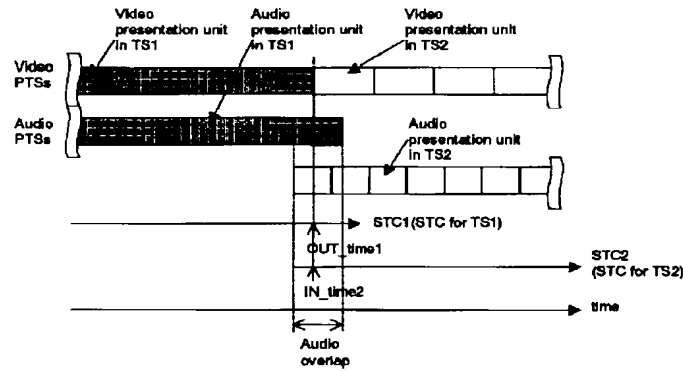


【図115】

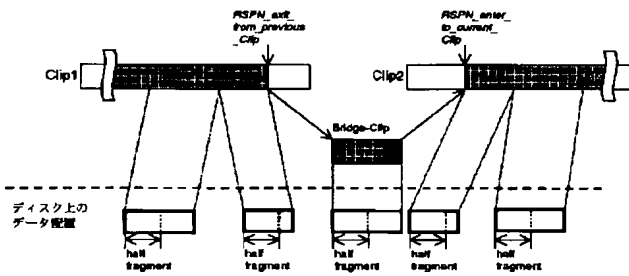
ClipMark

mark_type	mark_entry		representative_picture_entry	
	RSPN_ref_EP_start	offset_num_pictures	RSPN_ref_EP_start	offset_num_pictures
...	...	...	...	...
0x92(scene start)	A	M1	A	M2
0x94(CM start)	B	N1	B	N1
0x96(CM end)	C	N2	C	N2
...	...	...	...	...

【図101】

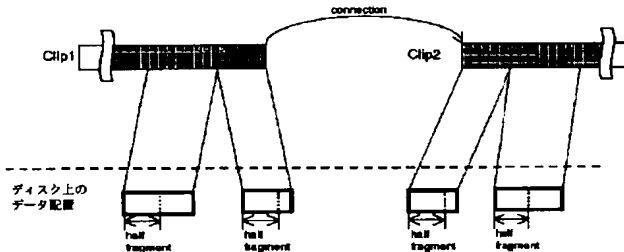


【図102】



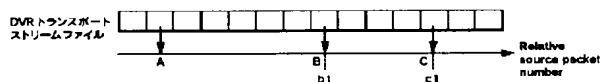
BridgeSequence を使用してシームレス接続をする場合の、データアロケーションの例

【図103】

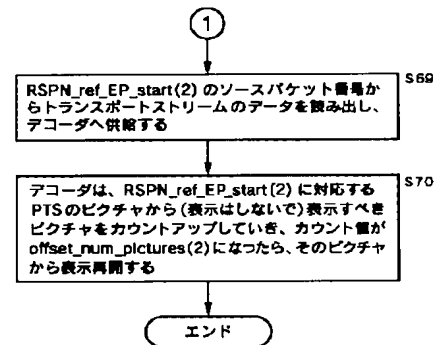


BridgeSequence を使用しないでシームレス接続をする場合の、データアロケーションの例

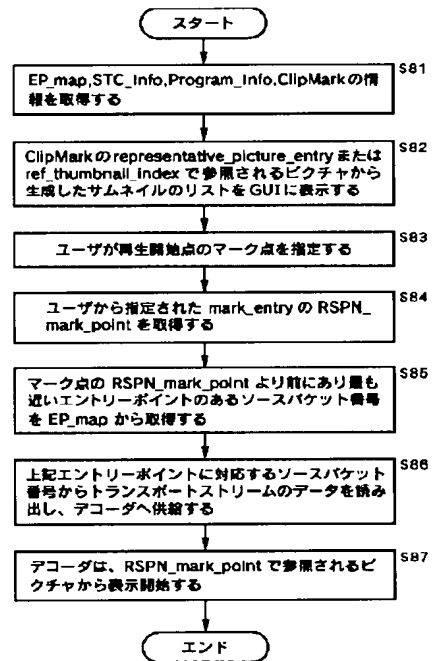
【図119】



【図117】



【図118】

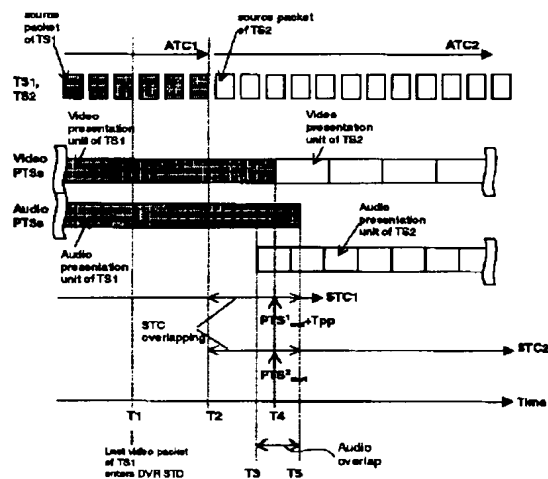


【図121】

ClipMark

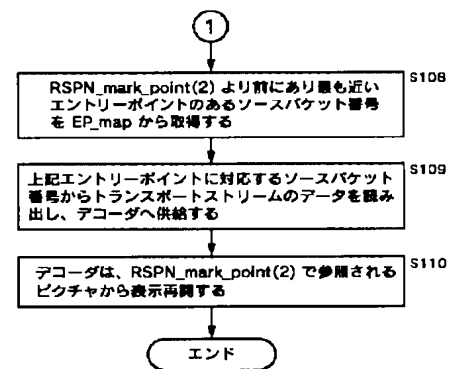
mark_type	ClipMark	
	mark_entry	representative_picture_entry
	RSPN_mark_point	RSPN_mark_point
...	...	...
0x92(scene start)	a 1	a 2
0x94(CM start)	b 1	b 1
0x95(CM end)	c 1	c 1
...	...	...

【図105】

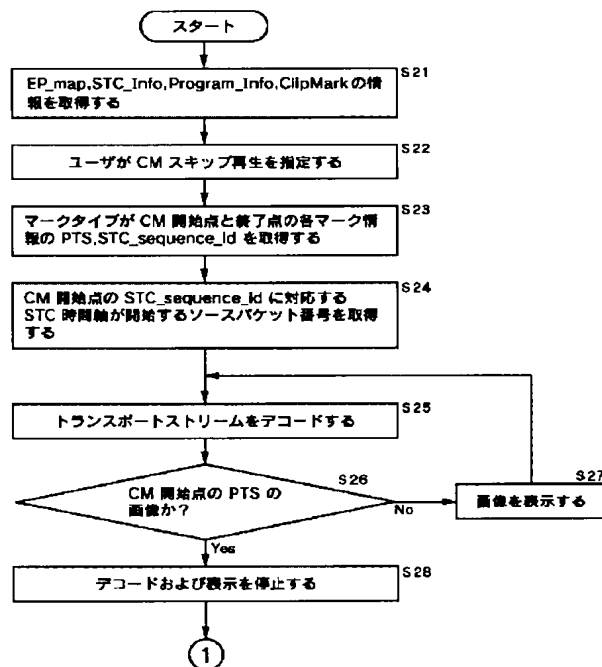


ある AV ストリーム(TS1)からそれにシームレスに接続された次の AV ストリーム(TS2)へと移る時のトランスポートパケットの入力、復号、表示のタイミングチャート

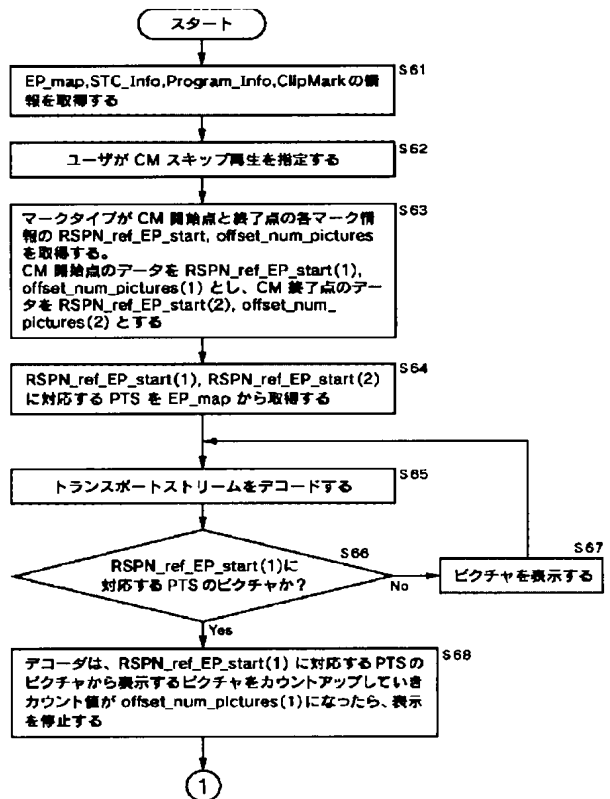
【図123】



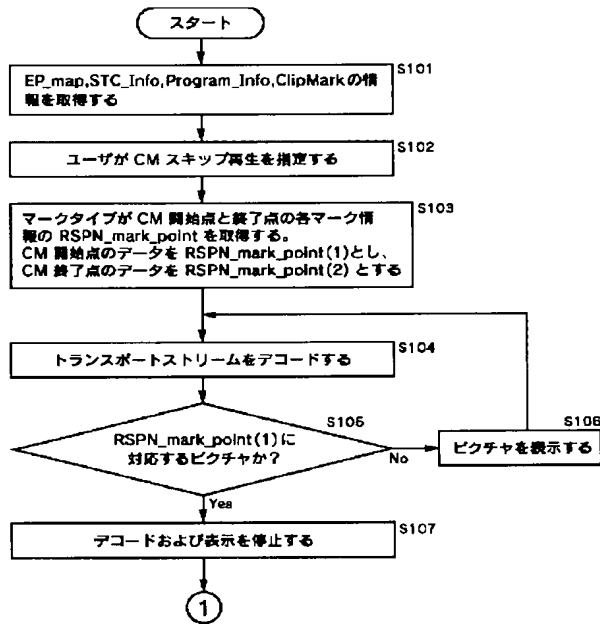
【図110】



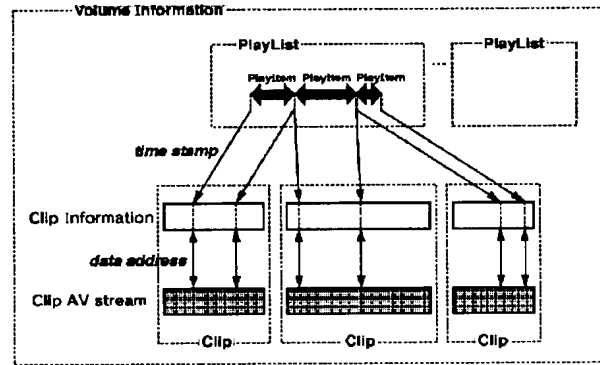
【図116】



【図122】

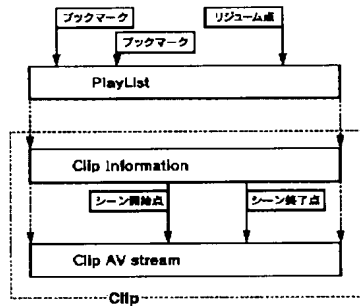


【図124】



【図125】

PlaylistMark  
タイムスタンプベースの指定  
(例) リジューム点、ブックマーク



ClipMark  
アドレスベースの指定  
(例) シーンスタート点、  
シーン終了点

【図126】

Syntax	No. of bits	Mnemonics
ClipMark0 {		
version_number	8*4	bsbtf
length	32	uimbsf
number_of_Clip_marks	16	uimbsf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bsbtf
mark_type	8	bsbtf
RSPN_mark	32	uimbsf
reserved	32	bsbtf
ref_thumbnail_index	16	uimbsf
}		
}		

【図128】

【図127】

Syntax	No. of bits	Mnemonics
ClipMark0 {		
version_number	8*4	bsbtf
length	32	uimbsf
number_of_Clip_marks	16	uimbsf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bsbtf
mark_type	8	bsbtf
RSPN_ref_EP_start	32	uimbsf
offset_num_pictures	32	uimbsf
ref_thumbnail_index	16	uimbsf
}		
}		

Syntax	No. of bits	Mnemonic
ClipInfo0 {		
Length	32	uimbsf
reserved_for_word_align	8	bsbtf
Clip_service_type	8	uimbsf
Clip_stream_type	8	uimbsf
reserved_for_word_align	6	bsbtf
transcode_mode_flag	1	bsbtf
time_controlled_flag	1	bsbtf
TS_average_rate	32	uimbsf
TS_recording_rate	32	uimbsf
reserved_for_DVRsystem_use	144	bsbtf
TS_type_Info_block()		
}		

ClipInfo0 のシンタクスの別例

【図129】

Syntax	No. of bits	Mnemonic
ProgramInfo() {		
length	32	uimsbf
reserved_for_word_align	8	bslbf
num_of_program_sequences	8	uimsbf
for (i=0; i<num_of_program_sequences; i++) {		
SPN_program_sequences_start	32	uimsbf
program_map_PID	16	bslbf
num_of_streams_in_ps	8	uimsbf
num_of_groups	8	uimsbf
for (stream_index=0; stream_index<num_of_streams_in_ps; stream_index++) {		
stream_PID	16	uimsbf
StreamCodingInfo()		
}		
} if (num_of_groups>1) {		
for (i=0; i<num_of_groups; i++) {		
num_of_streams_in_group	8	uimsbf
for (k=0; k<num_of_streams_in_group; k++) {		
stream_index	8	uimsbf
}		
} if (num_of_streams_in_group%2==0) {		
reserved_for_word_align	8	bslbf
}		
}		
}		

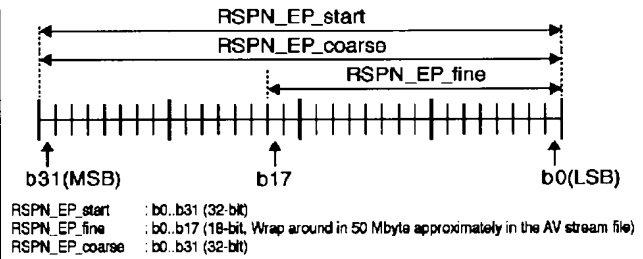
ProgramInfo() のシンタクスの別例

【図130】

Syntax	No. of bits	Mnemonic
StreamCodingInfo() {		
length	8	bslbf
stream_coding_type	8	uimsbf
if (stream_coding_type==0x02) {		
video_format	4	uimsbf
frame_rate	4	uimsbf
display_aspect_ratio	4	uimsbf
reserved_for_word_align	2	bslbf
cc_flag	1	uimsbf
original_video_format_flag	1	
if (original_video_format_flag==1) {		
original_video_format	4	uimsbf
original_display_aspect_ratio	4	uimsbf
reserved_for_word_align	8	bslbf
}		
} else if (stream_coding_type==0x03 // stream_coding_type==0x04 // stream_coding_type==0x0F // stream_coding_type==0x80 // stream_coding_type==0x81 //		
audio_presentation_type	4	uimsbf
sampling_frequency	4	uimsbf
reserved_for_word_align	8	bslbf
}		
}		

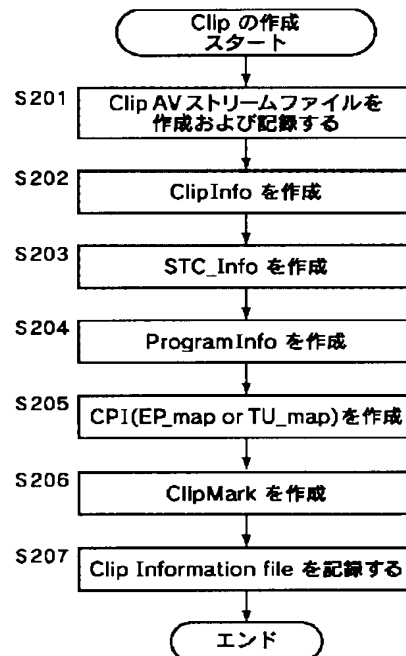
StreamCodingInfo() のシンタクス

【図134】



RSPN\_EP\_fine と RSPN\_EP\_coarse のフォーマットを説明する図

【図140】

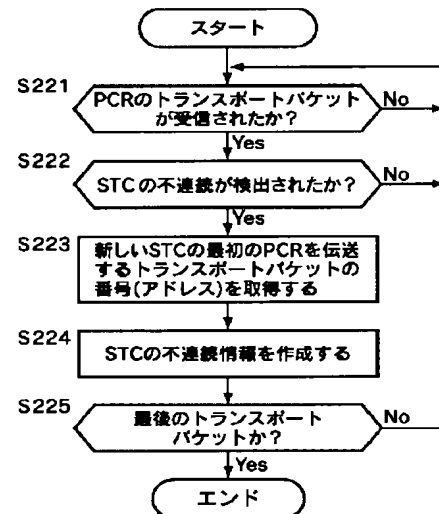


Clip AV ストリームファイルおよび Clip Information ファイルの作成

【図131】

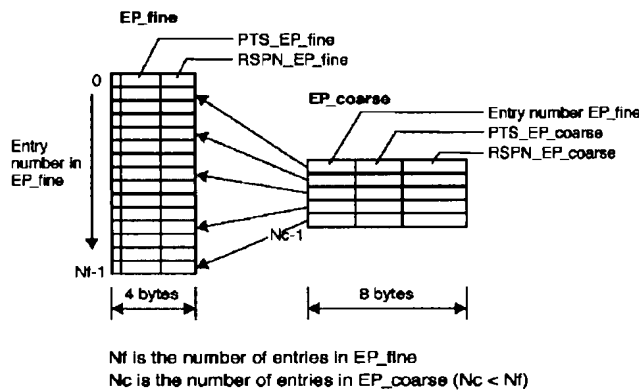
stream_coding_type	
stream_coding_type	Meaning
0x00 - 0x01	reserved for future use
0x02	MPEG-1 or MPEG-2 video stream
0x03	MPEG-1 audio
0x04	MPEG-2 multi-channel audio, backward compatible to MPEG-1
0x05	reserved for future use
0x06	Teletext defined in SESF or DVB or Subtitle defined in ISDB
0x07 - 0x09	reserved for future use
0x0A	ISO/IEC 13818-6 type A
0x0B	ISO/IEC 13818-6 type B
0x0C	ISO/IEC 13818-6 type C
0x0D	ISO/IEC 13818-6 type D
0x0E	reserved for future use
0x0F	MPEG-2 AAC audio with ADTS transport syntax
0x10 - 0x7F	reserved for future use
0x80	SESF LPCM audio
0x81	Dolby AC-3 audio
0x82 - 0xFF	reserved for future use

【図141】



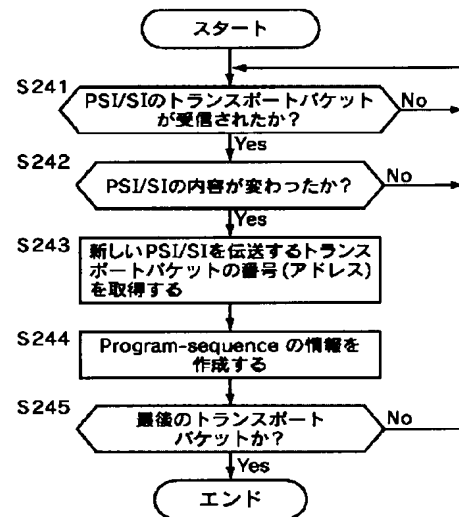
STC\_Info の作成の動作例を説明するフローチャート

【図132】



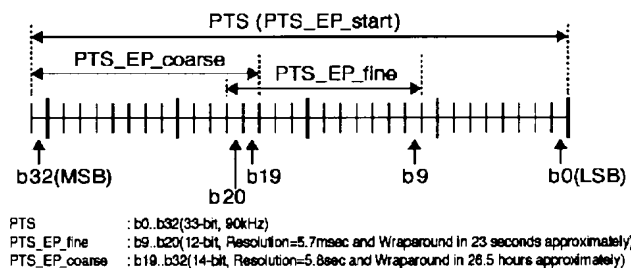
EP-fine と EP-coarse の関係を説明する図

【図142】



ProgramInfo の作成の動作例を説明するフローチャート

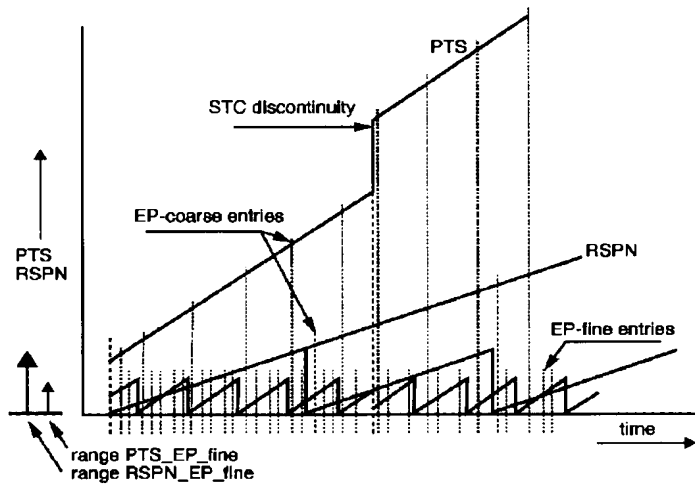
【図133】



PTS\_EP\_fine と PTS\_EP\_coarse のフォーマットを説明する図



【図135】



EP-coarseのエントリーとEP-fineのエントリーを説明する図

【図136】

Syntax	No. of bits	Mnemonic
EP_map(){		
reserved_for_word_align	8	bslbf
number_of_stream_PID_entries	8	uimsbf
for (k=0; k<number_of_stream_PID_entries; k++){		
stream_PID[k]	16	bslbf
reserved_for_word_align	10	bslbf
EP_stream_type[k]	4	uimsbf
num_EP_coarse_entries[k]	16	uimsbf
num_EP_fine_entries[k]	18	uimsbf
EP_map_for_one_stream_PID_start_address[k]	32	uimsbf
}		
for (l=0; l<X; l++){		
padding_word	16	bslbf
}		
for (k=0; k<number_of_stream_PID_entries; k++){		
EP_map_for_one_stream_PID (EP_stream_type[k],		
num_EP_coarse_entries[k],		
num_EP_fine_entries[k])		
for (l=0; l<Y[k]; l++){		
padding_word	16	bslbf
}		
}		
}		

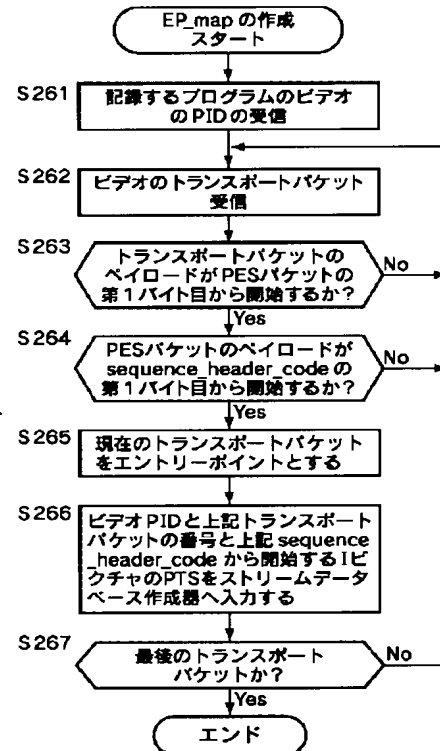
EP\_map のシンタックスの別例

【図137】

EP\_stream\_type values

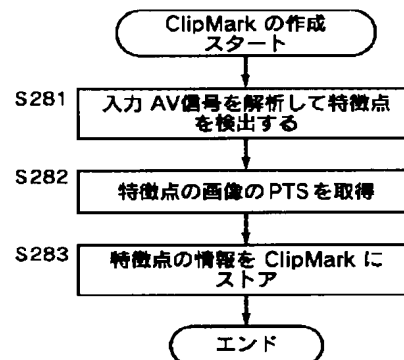
EP_stream_type	Meaning
0	video type1
1	video type2
2	audio
3-15	reserved for future use

【図143】



EP\_map の作成の動作例を説明するフローチャート

【図144】



アナログAV信号をエンコードして記録する場合における、図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図81のシンタックスであるときのClipMarkの作成方法を説明するフローチャート

【図138】

Syntax	No. of bits	Mnemonic
EP_map_for_one_stream_PID(EP_stream_type, Nc, Nf){		
EP_fine_table_start_address	32	uimsbf
for (i=0; i<Nc; i++){		
ref_to_EP_fine_id[i]	18	uimsbf
PTS_EP_coarse[i]	14	uimsbf
RSPN_EP_coarse[i]	32	uimsbf
}		
for (i=0; i<X; i++){		
padding_word	16	bslbf
}		
for (EP_fine_id = 0;		
EP_fine_id < Nf;		
EP_fine_id++){		
EP_video_type[EP_fine_id]	2	
PTS_EP_fine[EP_fine_id]	12	uimsbf
RSPN_EP_fine[EP_fine_id]	18	uimsbf
}		
}		

図136のEP\_mapのEP\_map\_for\_one\_stream\_PIDのシンタクス

【図139】

value	Meaning
0	エントリーポイントにあたるビデオアクセスユニットは、シーケンスヘッダから始まるIピクチャである。またそのIピクチャの前にGOPヘッダがあっても良い。SPN_EP_startは、そのアクセスユニットのシーケンスヘッダコードの第1バイト目を含むソースパケットのアドレスを示す。
1	エントリーポイントにあたるビデオアクセスユニットは、シーケンスヘッダから始まるPピクチャである。SPN_EP_startは、そのアクセスユニットのシーケンスヘッダコードの第1バイト目を含むソースパケットのアドレスを示す。
2	エントリーポイントにあたるビデオアクセスユニットは、シーケンスヘッダから開始しないIピクチャである。またそのIピクチャの前にGOPヘッダがあっても良い。 Iピクチャの前にGOPヘッダがある場合、SPN_EP_startは、そのアクセスユニットのグループ・スタート・コードの第1バイト目を含むソースパケットのアドレスを示す。 Iピクチャの前にGOPヘッダがない場合、SPN_EP_startは、そのアクセスユニットのグループ・スタート・コードの第1バイト目を含むソースパケットのアドレスを示す。
3	reserved for future use

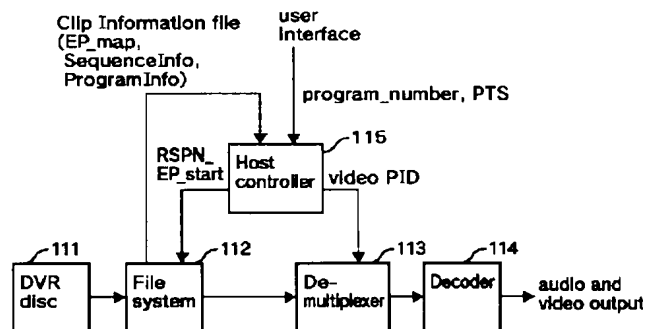
EP\_video\_type の値の意味

【図145】



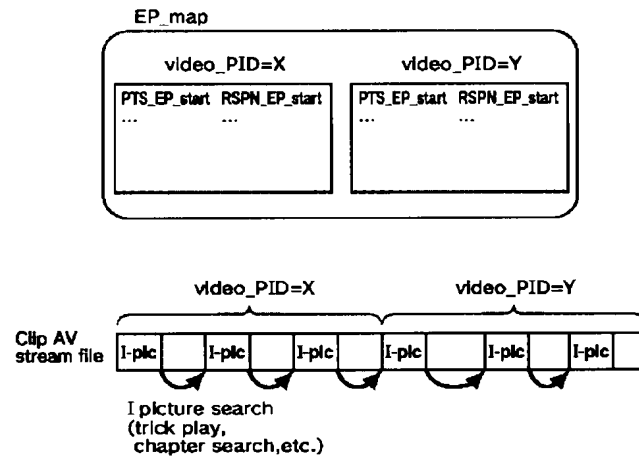
デジタルインタフェースから入力されたトランスポートストリームを記録する場合における、図75または図78のClipMarkのmark\_entry()/representative\_picture\_entry()が図81のシンタクスであるときのClipMarkの作成方法を説明するフローチャート

【図147】



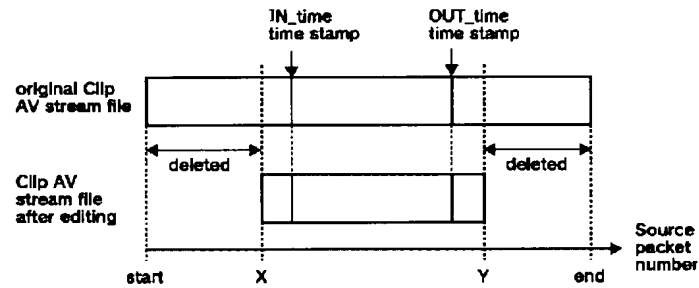
EP\_map を使用したIピクチャサーチのためのプレーヤモデル

【図146】



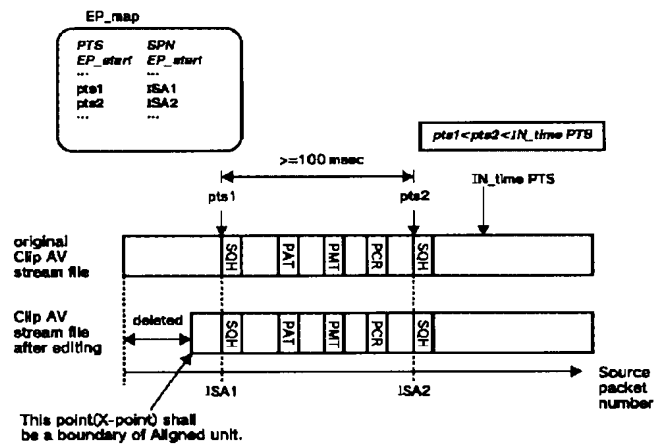
EP\_map を使う特殊再生を説明する図

【図148】



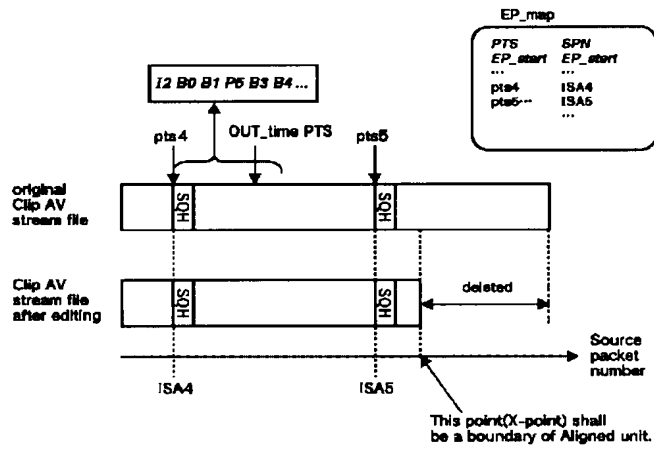
ミニマイズのおペレーションの例

【図149】



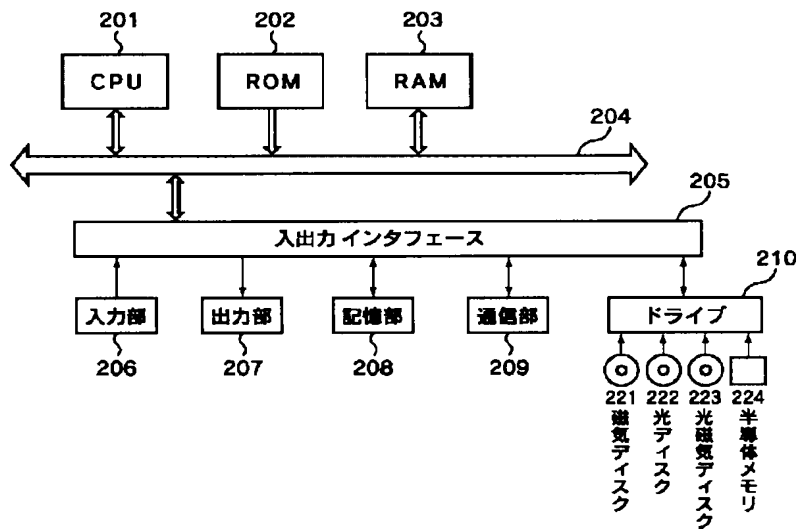
ミニマイズの時に IN\_time の前の不要なストリームデータを消去する例

【図150】



ミニマイズの時に OUT\_time の後ろの不要なストリームデータを消去する例

【図151】



フロントページの続き

F ターム(参考) 5C053 FA14 FA23 GB06 GB08 GB15  
 GB38 HA21 HA24 JA21 JA24  
 KA26 LA01 LA11  
 5D044 AB05 AB07 BC04 CC06 DE12  
 DE38 DE39 DE49 DE53 EF05  
 FG18 GK08  
 5D110 AA17 AA19 AA26 AA28 DA02  
 DA11 DA15 DA17 DA19 DB03  
 DB09 DC05 DC16